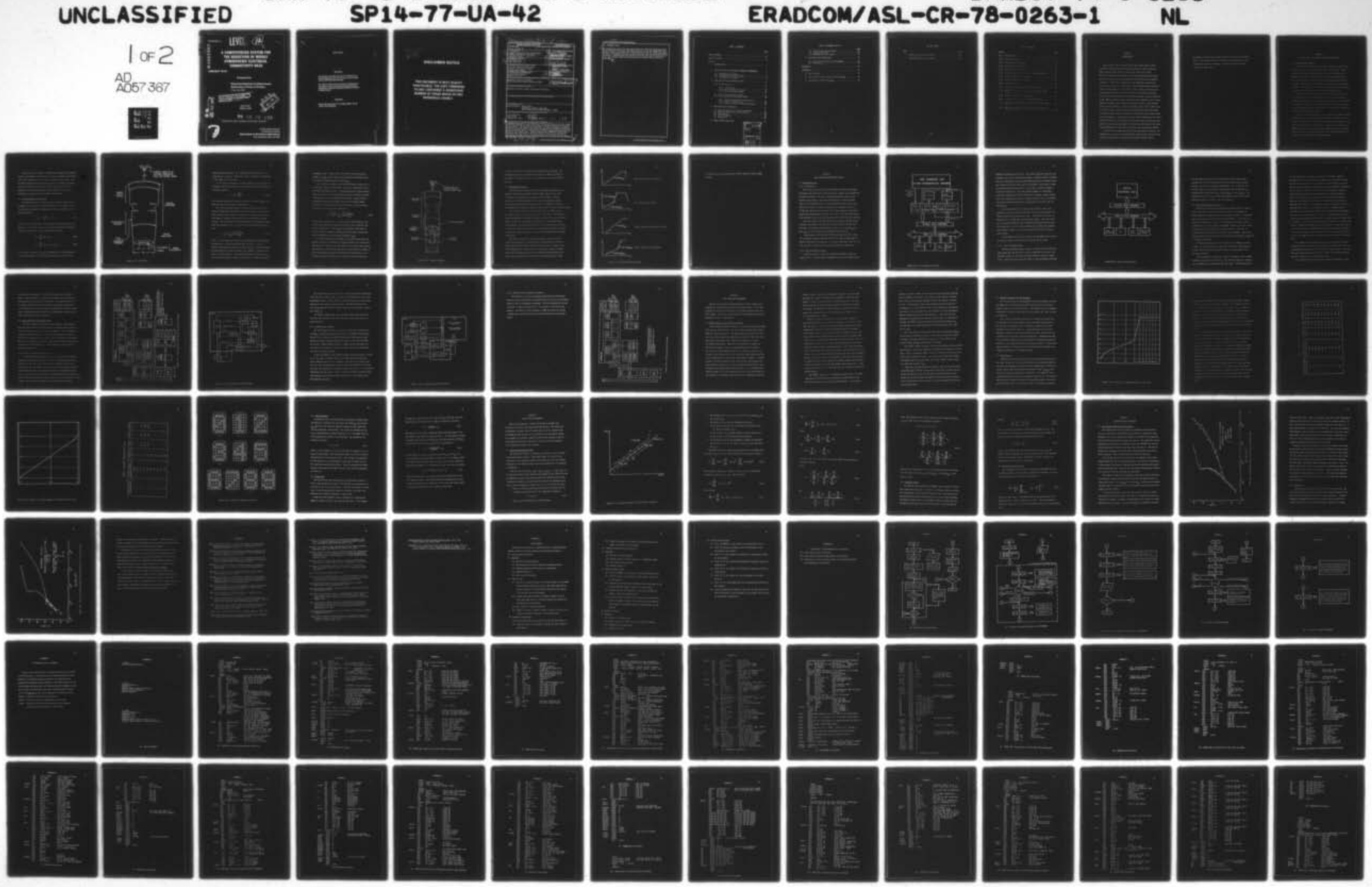
AD-A057 367 TEXAS UNIV AT EL PASO DEPT OF ELECTRICAL ENGINEERING F/G 4/1
A COMPUTERIZED SYSTEM FOR THE REDUCTION OF MIDDLE ATMOSPHERIC E--ETC(U)
JAN 78 S A SHIH , J D MITCHELL DAAD07-74-C-0263
UNCLASSIFIED SP14-77-UA-42 ERADCOM/ASL-CR-78-0263-1 NL

1 OF 2
AD
A057 367



ASL-CR-78-0263 - 1

LEVEL II

12

AD A057367

A COMPUTERIZED SYSTEM FOR THE REDUCTION OF MIDDLE ATMOSPHERIC ELECTRICAL CONDUCTIVITY DATA

JANUARY 1978

Prepared by

Electrical Engineering Department
University of Texas at El Paso
El Paso, Texas 79968

Under Contract
DAAD07-74-C-0263

US Army Electronics Research
and Development Command
Atmospheric Sciences Laboratory
White Sands Missile Range, N.M. 88002

NOTICES

Disclaimers

The findings in this report are not to be construed as an official Department of the Army position, unless so designated by other authorized documents.

The citation of trade names and names of manufacturers in this report is not to be construed as official Government indorsement or approval of commercial products or services referenced herein.

Disposition

Destroy this report when it is no longer needed. Do not return it to the originator.

(18) ERADCOM/ASL (19) CR-78-0263-1

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE — READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: ASL-CR-78-0263-1
2. GOVT ACCESSION NO.:
3. RECIPIENT'S CATALOG NUMBER:
4. TITLE (and Subtitle): (6) A COMPUTERIZED SYSTEM FOR THE REDUCTION OF MIDDLE ATMOSPHERIC ELECTRICAL CONDUCTIVITY DATA.
5. TYPE OF REPORT & PERIOD COVERED: (9) SPECIAL REP[redacted]T.
6. PERFORMING ORG. REPORT NUMBER: (14) SP14-77-UA-42
7. AUTHOR(s): (10) Shyue-wun A. Shih, John D. Mitchell
8. CONTRACT OR GRANT NUMBER(s): (15) DAAD07-74-C-0263
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Electrical Engineering Department, University of Texas at El Paso, El Paso, Texas 79968
10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: (16) DA Task 1L161102B53A SA2
11. CONTROLLING OFFICE NAME AND ADDRESS: US Army Electronics Research and Development Command, Adelphi, MD 20783
12. REPORT DATE: (11) Jan[redacted]78
13. NUMBER OF PAGES: 111
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office): Atmospheric Sciences Laboratory, White Sands Missile Range, New Mexico 88002
15. SECURITY CLASS. (of this report): UNCLASSIFIED (12) 118p.
15a. DECLASSIFICATION/DOWNGRADING SCHEDULE:

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

Contract Monitor: Robert Olsen, Atmospheric Sciences Laboratory, White Sands Missile Range, New Mexico 88002

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Ion conductivity	Stratosphere
Blunt probe	Mesosphere
Conductivity probe	Rocket-borne sensor

78 08 08 148

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

A computerized system for the reduction of middle atmospheric electrical conductivity data obtained by either blunt probes or Gerdien condensers is developed in this report. This particular system uses the Digital Equipment Corporation (DEC) PDP 11/10 minicomputer interfaced with the DEC LPS11 Laboratory Peripheral System, DEC LA36 writer II, Hewlett Packard HP3960 Instrumentation Recorder and Tektronix 603 Storage Scope. Assembly Language and FORTRAN IV programs were developed under the DEC RT-11 operating system to perform data digitizing, acquisition, storage, display, processing and finally printing out the results. → next page

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

408 579

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

20. ABSTRACT (cont)

cont.

The conductivity values from the computerized data reduction system were found to be consistent with those obtained by manually scaling the demodulated data waveforms from a strip chart (the method previously used for reducing the data). In fact, the computerized system is believed to be a more accurate and reliable technique. The method was also observed to enhance the range of sensitivity, i.e., the altitude region over which data can be reduced from a particular experiment.

TABLE OF CONTENTS

Page

LIST OF TABLES iii

LIST OF FIGURES. iv

Section

1. INTRODUCTION. 1

2. THE BLUNT PROBE AND GERDIEN CONDENSER EXPERIMENTS 3

   2.1 Experiment Description 3
   2.2 Current-Voltage Relationships 4
   2.3 Experimental Procedure 9

3. DATA REDUCTION MINICOMPUTER SYSTEM 12

   3.1 System Functions 12

      3.1.1 Introduction 12
      3.1.2 Data Acquisition System 12
      3.1.3 Data Processing System 14

   3.2 PDP 11/10 Minicomputer System 16
   3.3 LPS11 Laboratory Peripheral System 18

      3.3.1 LPSKW Programmable Real-Time Clock 18
      3.3.2 LPSVC Display Control 21
      3.3.3 LPSAD-12 Analog-to-Digital Converter 23

4. DATA REDUCTION PROGRAMMING 25

   4.1 Pulse Frequency and Timing Measurements 25
   4.2 Waveform Segmentation and Restorage 28
   4.3 Data Display 28
   4.4 Data Processing 35
   4.5 Timing Data 35

5. CURVE FITTING TECHNIQUES 37

TABLE OF CONTENTS (con't.)

	Page
5.1 Linear Least-Squares Method	37
5.2 Confidence Region	41
5.3 Residual Root Mean Square	42
6. DISCUSSION AND CONCLUSIONS	43
6.1 Electrical Conductivity Measurements	43
6.2 Conclusions	45
REFERENCES	48
APPENDICES	51
A. User's Manual	51
B. Flow Chart of Minicomputer (RT-11) Programs	54
C. Minicomputer (RT-11) Programs	60

LIST OF TABLES

Table		Page
1.	Original Scalings on the Scope	31
2.	Expanded Scalings on the Scope	33

LIST OF FIGURES

Figures		Page
(2-1)	Blunt Probe	5
(2-2)	Gerdien Condenser	8
(2-3)	Representative Waveforms	10
(3-1)	Data Acquisition System	13
(3-2)	Data Processing System	15
(3-3)	LPS11 Laboratory Peripheral System Block Diagram	19
(3-4)	Real-Time Clock Block Diagram	20
(3-5)	Display Control Block Diagram	22
(3-6)	A/D Converter Block	24
(4-1)	Display of a Complete Waveform on the Scope	29
(4-2)	Display of a Specific Region of a Waveform on the Scope	32
(4-3)	Display of Numerical Characters	34
(5-1)	Confidence Region of Least-Squares Estimates	38
(6-1)	Parachute-Borne Blunt Probe Electrical Conductivity Data	44
(6-2)	Balloon and Parachute-Borne Blunt Probe Electrical Conductivity Data	46

SECTION 1
INTRODUCTION

Blunt probes [Hale and Hoult (1965); Hale (1967); Hale, Hoult and Baker (1968)] and Gerdien condensers [Pedersen (1964); Rose and Widdel (1972); Conley (1974); Croskey, Hale and Leiden (1977); Mitchell, Sagar and Olsen (1977)] are presently being flown on rocket and balloon systems to measure electrical conductivity in the middle atmosphere. The Gerdien condenser has the additional capability of being able to measure ion mobility and charge number density [Pedersen (1964); Croskey (1976); Sagar (1976)]. Both of these experiments are being utilized to study ionization process in the middle atmosphere. In particular, such phenomena as midlatitude sunrise condition [Mitchell et al. (1977)] and the D-region "winter anomaly" [Mitchell, Hale, Olsen, Randhawa and Rubio (1972); Mitchell and Hale (1973)], a solar eclipse [Baker and Hale (1970)], a polar cap absorption event [Hale (1974)] and the high-latitude, middle atmosphere during geomagnetically disturbed conditions [Olsen, Mitchell and Croskey (1976)] have been studied using rocket instruments. A balloon-borne blunt probe experiment on the recent STRATCOM flights has also proven useful in studying the temperature dependence and altitude dependence of electrical conductivity in the stratosphere [Mitchell and Hale (1973); Mitchell, Hale and Croskey (1977)].

With the recent miniaturization of the blunt probe to make it compatible with the super Loki meteorological rocket system, and with this particular instrument now commercially available [Olsen (1977)],

the number of blunt probe rocket flights has increased appreciably. This, in turn, has resulted in the need for computerized data processing and reduction techniques to improve both the speed and accuracy of these tasks.

SECTION 2

THE BLUNT PROBE AND GERDIEN CONDENSER EXPERIMENTS

2.1 Experiment Description

The subsonic blunt probe [Hale *et al.* (1965); Hale (1967)] and Gerdien condenser [Pedersen (1964); Croskey (1976); Sagar (1976)] experiments measure electrical conductivity and in addition, the Gerdien condenser measures ion mobility and charge number density. When launched using a rocket, the payload separates from the motor at apogee (nominally at 70 to 80 km) and descends to the ground on a parachute. The data are telemetered in flight back to the receiving station where the information is recorded on magnetic tape.

The blunt probe experiment was initially designed in the mid 1960's [Hale *et al.* (1965); Hale (1967)]. The theory of charged particle collection for this instrument indicates that to first order, the collection current is neither dependent on the descent velocity nor the pendulum motion of the payload as it descends on a parachute [Hale *et al.* (1965); Hoult (1965)].

With the recent development of more stabilized parachute systems, the possibility of a subsonic Gerdien condenser experiment for measuring ion mobility and charge number density (both of which are flow dependent parameters) is now feasible. Thus, subsonic Gerdien condenser experiments for flying on such stabilized parachute systems have been recently developed [Farrokh (1975); Croskey (1976); Sagar (1976)] to measure these electrical parameters.

The particular instruments for which the computer data reduction scheme in this report was developed are flown on standard meteorological rocket systems such as the Arcas and super Loki rockets. The 1680 MHz transmitter and modulation system is compatible with those used by the Meteorological Rocket Network (MRN), thus making it possible to launch the instruments at any MRN rocket range [The Meteorological Rocket Network Document 11-64 (1965)].

2.2 Current-Voltage Relationships

The blunt probe uses a circular planar collector geometry for charged particle collection (see Figure (2-1)). The current of collected charged particles is described by the equation [Hale (1967); Mitchell (1973)]:

$$|I_\pm| = \frac{2\,r^2}{R}\,\sigma_\pm\,|V| \tag{2-1}$$

where r and R are the radii of the collector and the outside of the guard ring, respectively, and $\sigma_\pm$ is either the positive or negative electrical conductivity which is defined as follows:

$$\sigma_+ = \sum_i N_{i+}\,e\,\mu_{i+} \tag{2-2}$$

$$\sigma_- = \sum_i N_{i-}\,e\,\mu_{i-} + N_e\,e\,\mu_e \tag{2-3}$$

In the above expressions, N_{i+} (N_{i-}) represents the concentration of positive (negative) ions of the ith species and μ_{i+} (μ_{i-}) is its

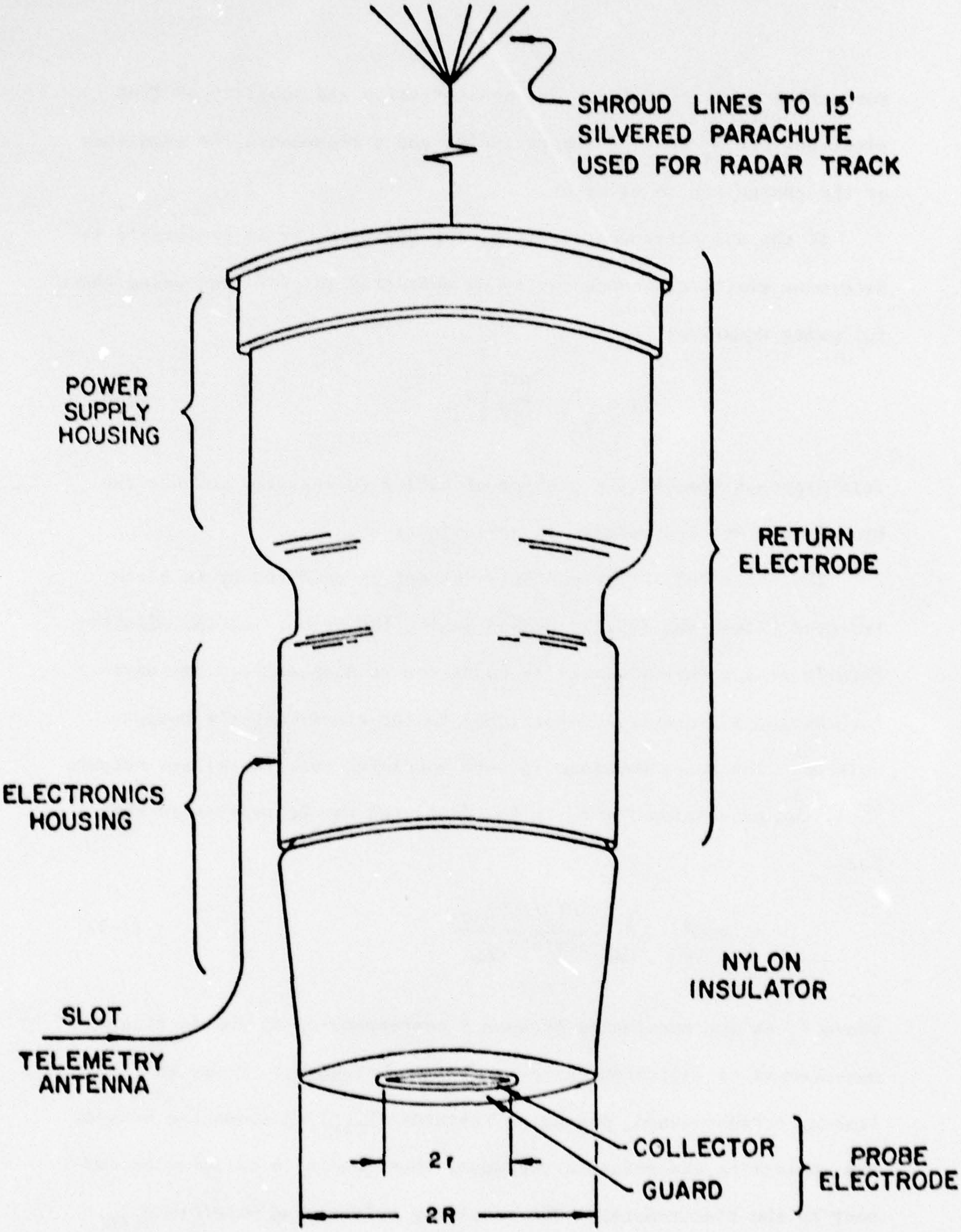


Figure (2-1) Blunt Probe

respective mobility value. The concentration and mobility of free electrons are N_e and μ_e, respectively, and e represents the magnitude of the charge for an electron.

If the collector voltage is a ramp function, it is preferable to determine electrical conductivity by measuring $(dI_{\pm}/dV)$ and using the following equation:

$$\sigma_{\pm} = \frac{R}{2r^2} \left|\frac{dI_{\pm}}{dV}\right| \tag{2-4}$$

This approach removes the problem of having to actually measure the probe's current and voltage in determining $\sigma_{\pm}$.

The collected charge particle current is measured by an electrometer [Zimmerman (1971)] housed inside the probe, and the electrometer's analog output signal is converted to a negative pulse waveform having a frequency proportional to the electrometer's output voltage. The pulse waveform in turn modulates the transmitter output. Thus, the expression for $\sigma_{\pm}$ in Eq. (2-4) can now be written in the form

$$\sigma_{\pm} = \frac{R}{2r^2} \frac{1}{R_{CAL}} \frac{(df_{\pm}/dt)_{DATA}}{(df/dt)_{CAL}} \tag{2-5}$$

where $f_{\pm}$ is the modulation frequency corresponding to the in-flight measurement of collected charge particle current $I_{\pm}$. Prior to launch, a high-valued, precision resistor (R_{CAL}) is connected between the collecting and return electrodes, thus feeding a calibration current to the electrometer. The resulting telemetered waveform f_{CAL}, is received through the telemetry system prior to flight and recorded

on magnetic tape. In Eq. (2-5), the value of this calibration resistor is R_{CAL} and the slope of the modulated calibration waveform during the sweep portion is $(df/dt)_{CAL}$.

For a Gerdien condenser, the collector geometry consists of a cylindrical collector and an outer, concentric cylindrical return electrode (see Figure (2-2)). A voltage waveform is swept between these two electrodes and the resulting current of charge particles collected on the inner electrodes is measured. Using a similar electronics system and preflight calibration procedure [Sagar (1976)] as for the blunt probe, the expression $\sigma_{\pm}$ in the Gerdien condenser's linear region of operation is

$$\sigma_{\pm} = \frac{\ln(r_o/r_i)}{2\pi\ell} \frac{1}{R_{CAL}} \frac{(df_{\pm}/dt)_{DATA}}{(df/dt)_{CAL}} \tag{2-6}$$

In the above expression, r_o, r_i and ℓ are the inner radius of the return electrode and the radius and length of the collecting electrode, respectively. Again, R_{CAL} is the resistor value used in parallel with the condenser to generate the preflight calibration current resulting in the modulating frequency f_{CAL}.

If the flow through the aspirator can be determined, then the Gerdien condenser's cylindrical electrode geometry also affords the opportunity for measuring the ion mobility and charge number density. The reduction of the Gerdien condenser's current-voltage response to obtain the mobility information requires the determination of the voltages at which the different ion mobility groups are entirely

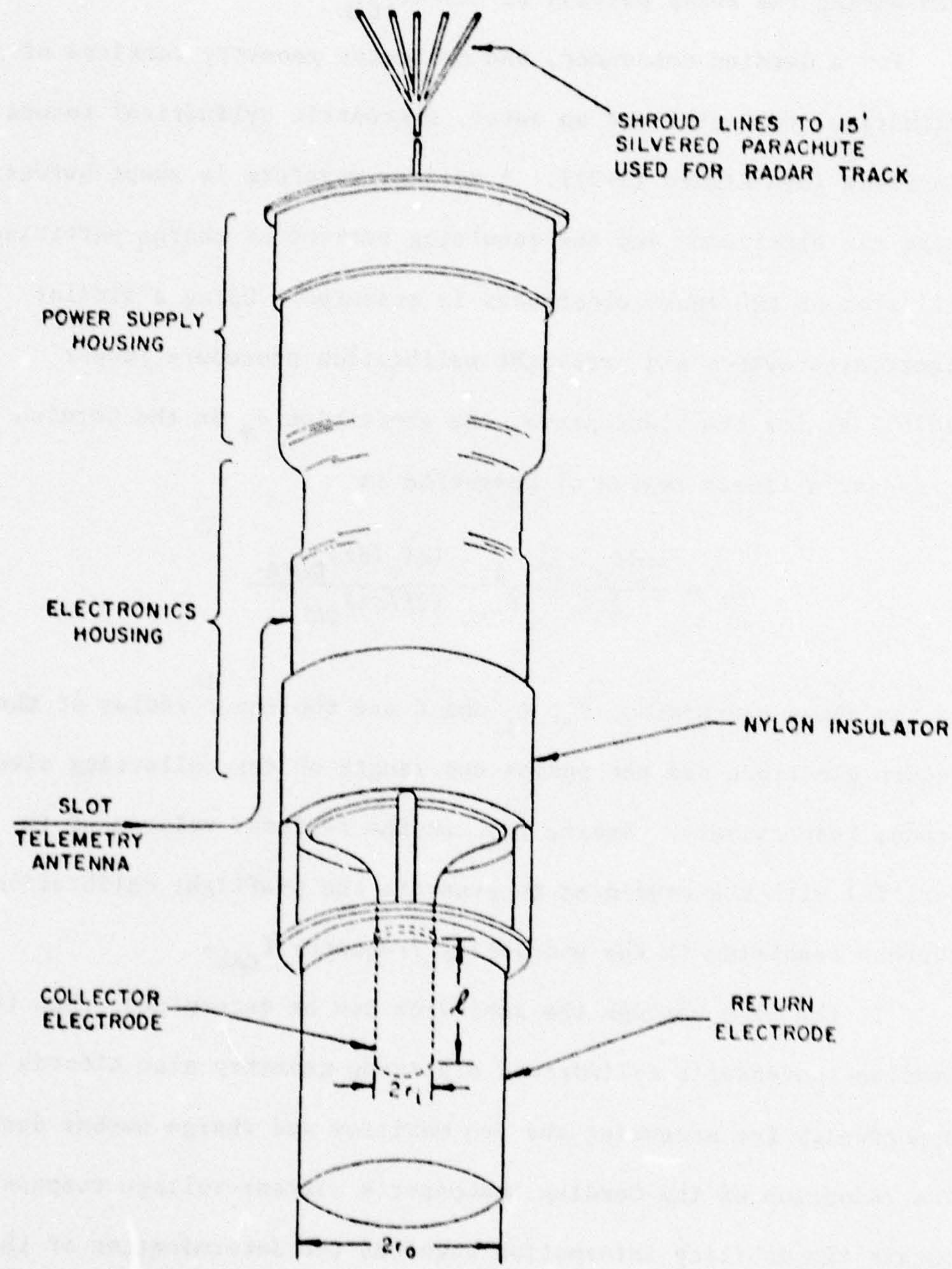


Figure (2-2) Gerdien Condenser

collected out of the air sample passing through the aspirator. The determination of these voltage values requires a further analysis of the probe's I-V response and was not considered in the scope of this research.

2.3 Experimental Procedure

As mentioned previously, the rocket experiments are usually conducted at Meteorological Rocket Network (MRN) launch sites. Prior to launch, the instrument is operated in the preflight calibration mode as discussed earlier. The resulting transmitted signal is received and recorded on magnetic tape. An example of this waveform showing the calibration frequency versus time is given in Figure (2-3).

The telemetry system used for obtaining the in-flight data is the same as that used during the preflight calibration. In addition, timing information is recorded on another channel of the tape recorder. While the probe is descending on a parachute, a ground-based radar system measures the position and velocity of the instrument as a function of time and thus the electrical conductivity data, which are also recorded as a function of time, can later be determined as a function of altitude.

The actual reduction of the data waveforms occurs at a later time in the laboratory. Representative data waveforms for both the blunt probe and the Gerdien condenser are also shown in Figure (2-3). As discussed earlier, the electrical conductivity values are proportional to the designated slopes of the modulated data waveforms and thus, the data reduction procedure involves determining the slopes

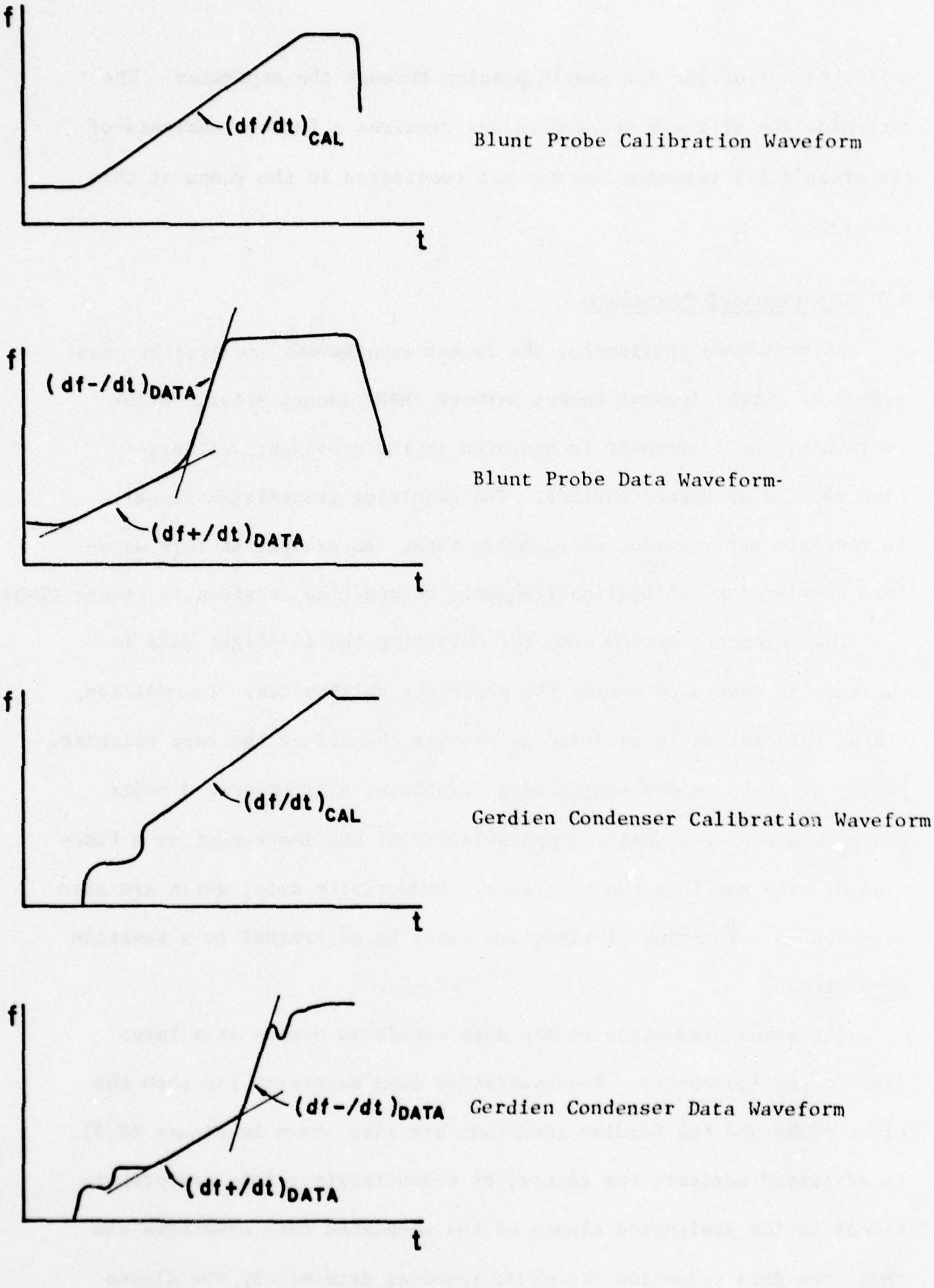


Figure (2-3) Representative Waveforms

of these particular waveforms [Hale (1967); Mitchell (1973); Sagar (1976)].

SECTION 3

DATA REDUCTION MINICOMPUTER SYSTEM

3.1 System Functions

3.1.1 Introduction

The overall minicomputer system performs two general functions in reducing the electrical conductivity data, namely, data acquisition and data processing. The data acquisition procedure involves the transfer and storage of data from the originally recorded magnetic tape to a DEC RK11/RK05 disk. In transferring the data, which initially are a series of negative pulses in the frequency range of 0 to 200 pps, the time period between the leading edges of consecutive pulses are measured using the Real-Time Clock of the DEC LPS11 Laboratory Peripheral System and the digitized values are stored on the disk.

The processing of the data involves such tasks as waveform segmentation and display, the removal of spurious noise from the data waveform, and actually determining the slope, i.e., $(df_{\pm}/dt)_{DATA}$ or $(df/dt)_{CAL}$ of the waveform for a designated time interval.

A discussion of the DEC PDP 11/10 minicomputer and LPS11 Laboratory Peripheral System, which are inherent to both of these data reduction functions, will be discussed in later sections of this section. Also, a user's manual of this system is presented in Appendix A.

3.1.2 Data Acquisition System

The block diagram of the data acquisition system is shown in Figure (3-1). The data inputs originate from the tape transport unit

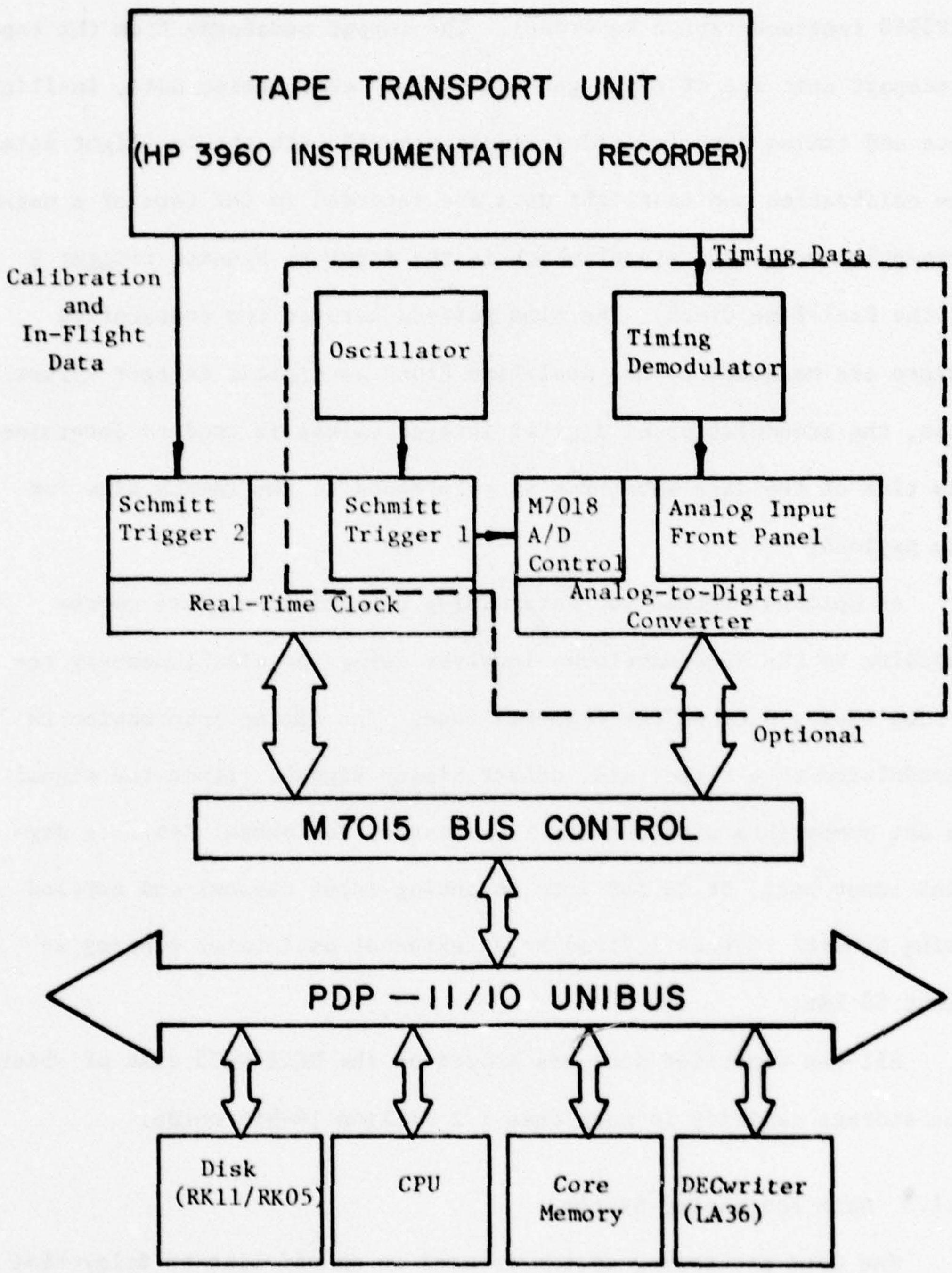


Figure (3-1) Data Acquisition System

(HP3960 Instrumentation Recorder). The output waveforms from the tape transport unit are of three general types: calibration data, in-flight data and timing data (recorded simultaneously with the in-flight data). The calibration and in-flight data are recorded in the form of a negative-pulse modulated signal which is the input to Schmitt trigger 2 of the Real-Time Clock. The time periods between two consecutive pulses are measured by the Real-Time Clock as digital integer values. Also, the accumulation of digital integer values is used to determine the time of the data waveforms as referenced to the launch time for the payload.

An optional method for determining the relative times corresponding to the data waveforms involves using the simultaneously recorded timing data on the magnetic tape. The timing information is demodulated to a fixed rate, offset binary signal. Since the signal is not compatible with the LPS11 Laboratory Peripheral System's digital input port, it is fed into an analog input channel and sampled using Schmitt trigger 1 fired by an external oscillator running at about 30 kHz.

All the digitized data are stored on the RK11/RK05 disk of which the storage capacity is more than 1.2 million 16-bit words.

3.1.3 Data Processing System

The data processing system is used in an off-line or delay-time sense after data are initially stored as described in Section 3.1.2. The block diagram of the data processing system is shown in Figure (3-2). The Central Processor Unit (CPU) of the minicomputer performs

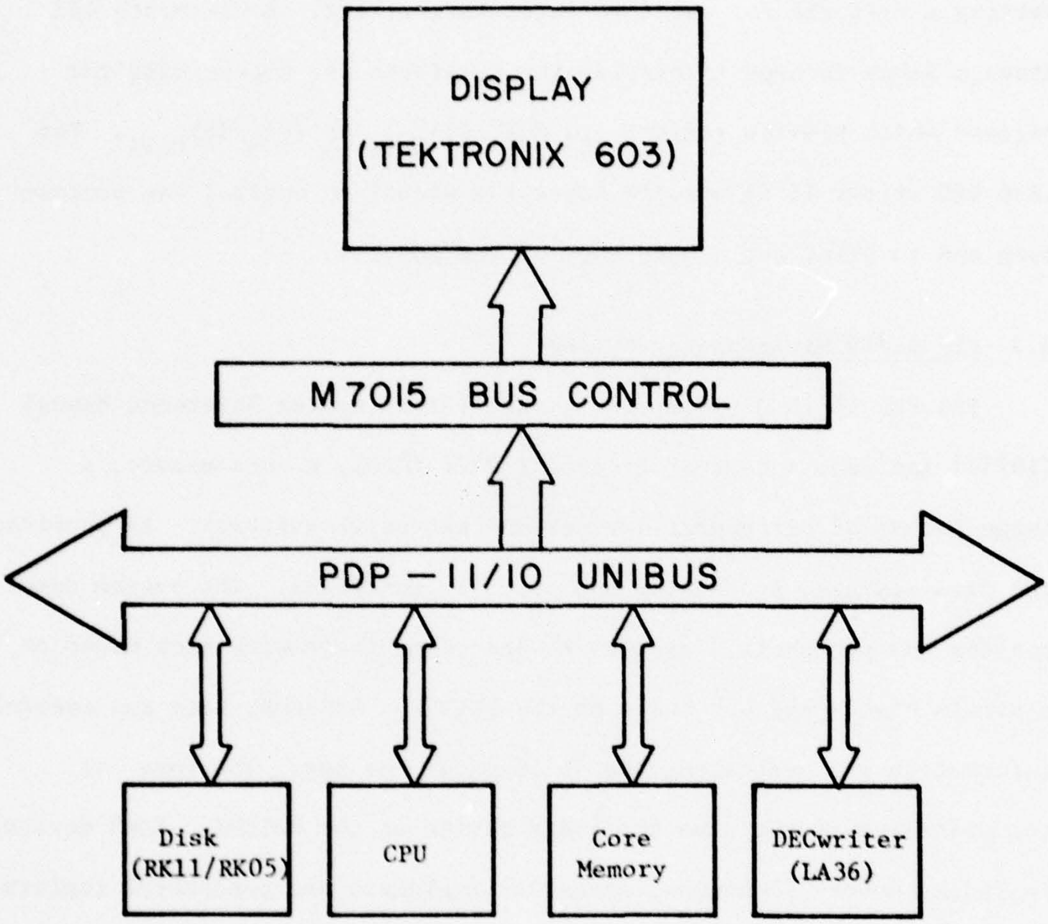


Figure (3-2) Data Processing System

all the numerical functions and calculations such as determining a straight line fit to the data waveform by the least-squares method, setting a criteria for waveform segmentation, etc. A Tektronix 603 Storage Scope is used to display the waveforms for determining the regions which provide $(df/dt)_{CAL}$, $(df_+/dt)_{DATA}$ or $(df_-/dt)_{DATA}$. The LA36 DEC writer II is used to input the signal to control the processing and to print out a hard copy of the results.

3.2 PDP 11/10 Minicomputer System

The PDP 11/10 minicomputer system [RT-11 System Reference Manual (1975)] includes a Central Processor Unit (CPU), a core memory, a large number of peripheral devices and extensive software. It provides the data storage, processing and printout functions. The system components and peripherals connect to and communicate with each other on a single high-speed bus known as the UNIBUS. Address, data and control information are sent along the 56 lines of the bus. The form of communication is the same for every device on the UNIBUS. Each device, including memory locations, processor registers and peripheral registers, is assigned an address.

The Central Processor Unit is connected to the UNIBUS as a subsystem. It performs arithmetic and logic operations, instruction decoding, and data transfers directly between the input/output (I/O) devices and memory.

The core memory is viewed as a series of locations, with a number (address) assigned to each location. The PDP 11/10 memory is designed to accommodate both 16-bit words and 8-bit bytes. A 16-bit word used

for byte addressing can address a maximum of 32K words. However, addresses from 0 to 777_8 and the top $4,096_{10}$ word locations have been reserved by the system for the interrupt and trap handling, processor stacks, general registers, and peripheral devices registers, and therefore, a maximum of 28K of core are left to be programmed. However, only 16K words of core have been implemented in this minicomputer system. The amount of mass storage in the minicomputer is another important consideration in this system. The PDP 11/10 system has a RK11/RK05 disk and two TA11 cassette tape drives which can be used for immediate mass storage. The RK11/RK05 disk has a maximum storage capacity of over 1.2 million 16-bit words per disk and a data transfer speed of 11.1 microseconds (μs) per word. The RK11/RK05 disk is fast enough and has enough storage capacity for about one hour of flight time. For example, if the average frequency of the flight data is 100 Hz, this requires 360,000 16 bit-words of storage capacity if every data point is to be stored. Thus, if 10,000 data points are transferred from memory location to the disk, it only takes 0.111 second. Also, if the direct memory access (DMA) operation is used, it takes less than 1 millisecond (ms).

The cassette drive system is too slow to be used while the data acquisition system is running since there is not enough time for tape positioning and data transfer from memory to cassette tape. After the data have been processed and stored on a disk, the cassettes could be used as a cheaper form of storage for backup.

The communication between the user and the minicomputer is also

an important consideration. The PDP 11/10 system has a LA36 DEC writer II which is used as a console for inputting the control data and printing out the necessary information. The LA36 DEC writer II is loaded with many practical functional and operator features such as 30 character per second throughput (accomplished by a 60 Hz catchup mode), infinitely variable vertical forms adjustment, variable forms width, and multi-part forms capability.

3.3 LPS11 Laboratory Peripheral System

The LPS11 Laboratory Peripheral System includes a programmable Real-Time Clock with two Schmitt triggers, a Display Control with two 12-bit D/A converters, and a 12-bit A/D converter. It is a high performance, modular and real-time subsystem that interfaces with the PDP 11/10 minicomputer via the UNIBUS. The flexibility of the system makes it well suited for a variety of applications such as data collection, monitoring and reduction. A block diagram of this system is shown in Figure (3-3).

3.3.1 LPSKW Programmable Real-Time Clock

The LPSKW Real-Time Clock offers several methods for accurately measuring and counting intervals or events. A block diagram is shown in Figure (3-4). The clock can be used to synchronize the central processor to external events, count external events, measure intervals of time between events, and provide interrupts at programmable intervals. It can also be used to start the analog-to-digital converter by means of the overflow from the clock counter or by the firing of a Schmitt trigger. Many of these operations can be performed concurrently.

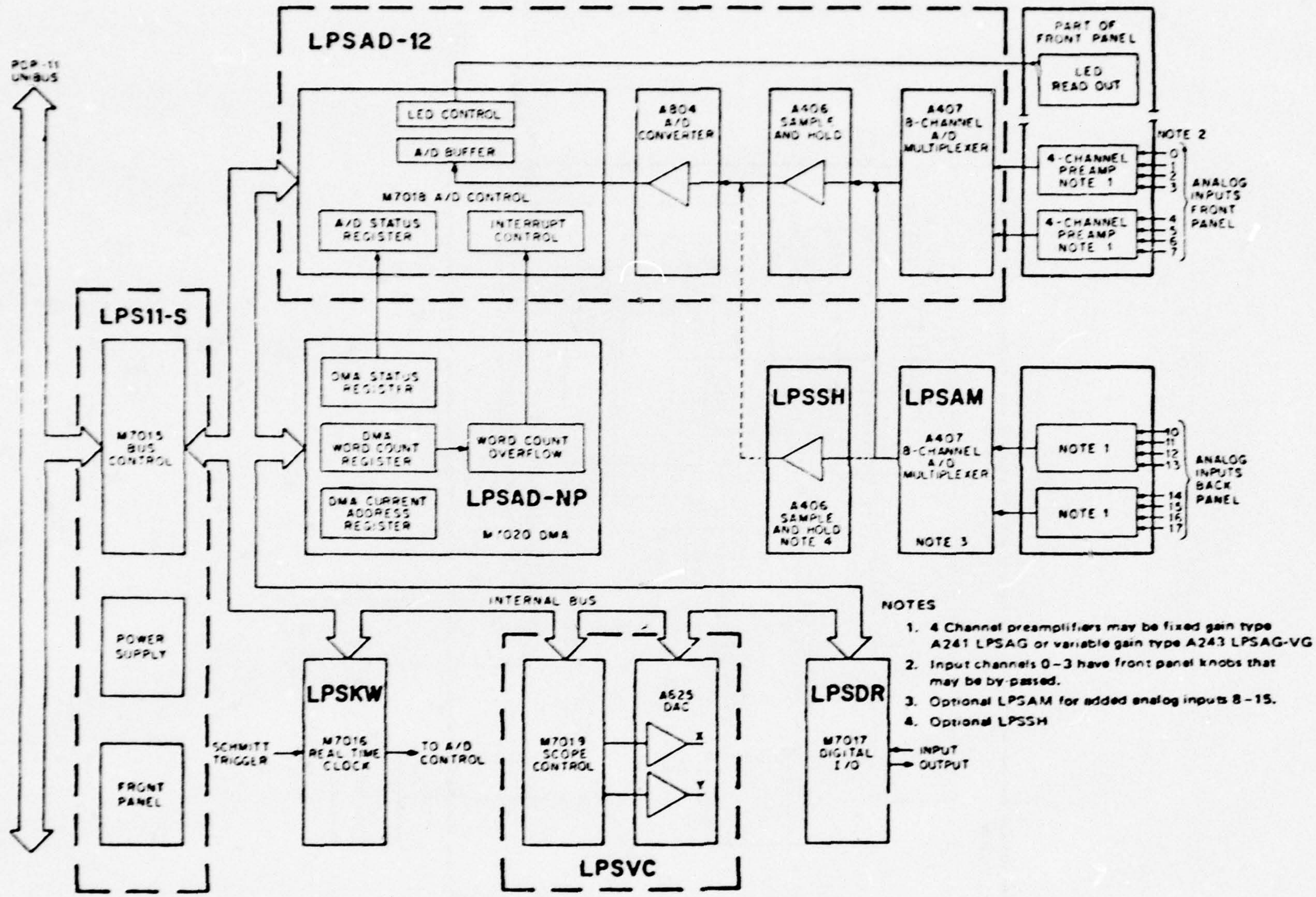


Figure (3-3) LPS11 Laboratory Peripheral System Block Diagram

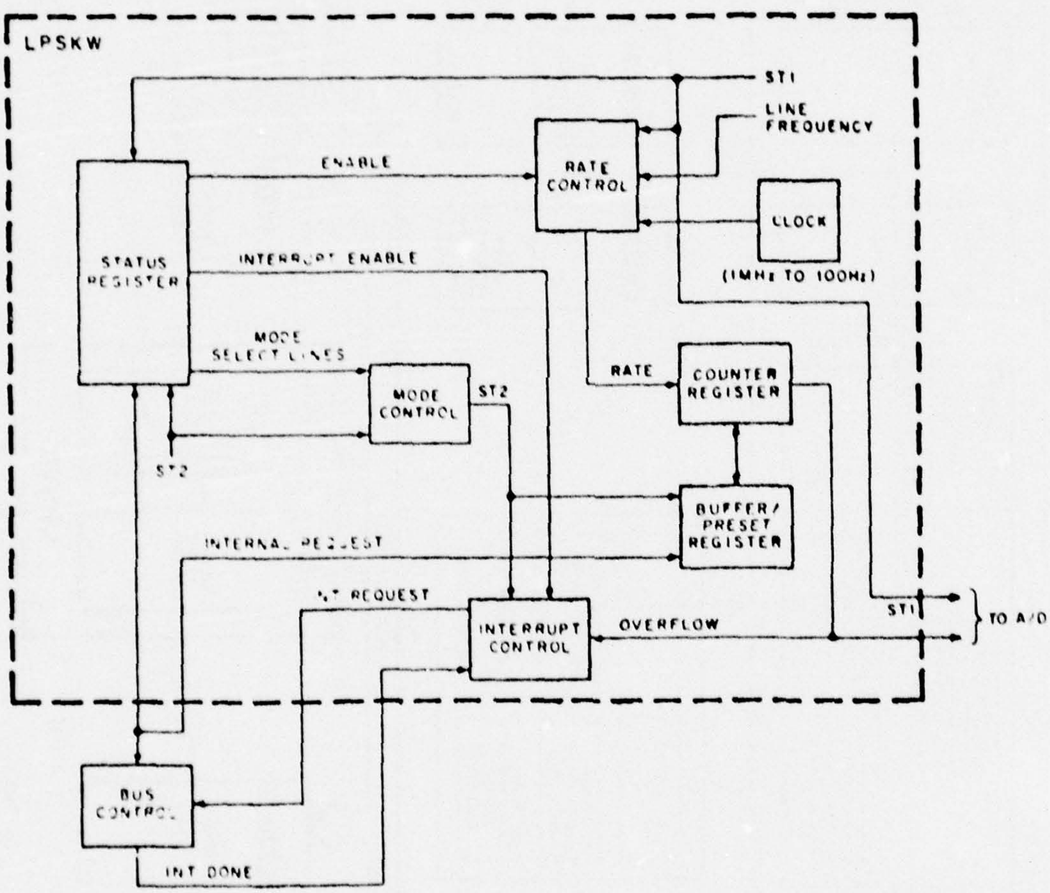


Figure (3-4) Real-Time Clock Block Diagram

The clock will generate one of five crystal-controlled frequencies: 1 MHz, 100 kHz, 10 kHz, 1 kHz or 100 Hz, and operate in any one of four programmable modes: single interval, repeated interval, external event timing and event counting from zero base. The Real-Time Clock may also use an external (Schmitt trigger) input or a line frequency input as a time base.

Two Schmitt triggers which are included with the Real-Time Clock can start and read the clock, start the A/D converter and cause program interrupts.

3.3.2 LPSVC Display Control

The LPSVC Display Control is used to display data in the form of a $4{,}096_{10} \times 4{,}096_{10}$ dot array on the scope. The Display Control (see Figure (3-5)) consists of an M7019 Scope Control Module and an A625 Digital-to-Analog converter Module which must be used with the M7015 Bus Control. Under program control, a bright dot may be produced at any point in this array, or a series of these dots may be programmed to produce a graphical output.

Output operations of the Display Control, which may output to either an X/Y recorder or a display unit, are accomplished by loading the status register and the X or Y register. Through use of status register bits, the Display Control, which operates the Tektronix 603 Storage Scope has the capability of intensifying the contents of X or Y registers, indicating when the scope is ready for intensification, providing erase, write-through, and non-stop control functions for the storage scope, and enabling interrupts.

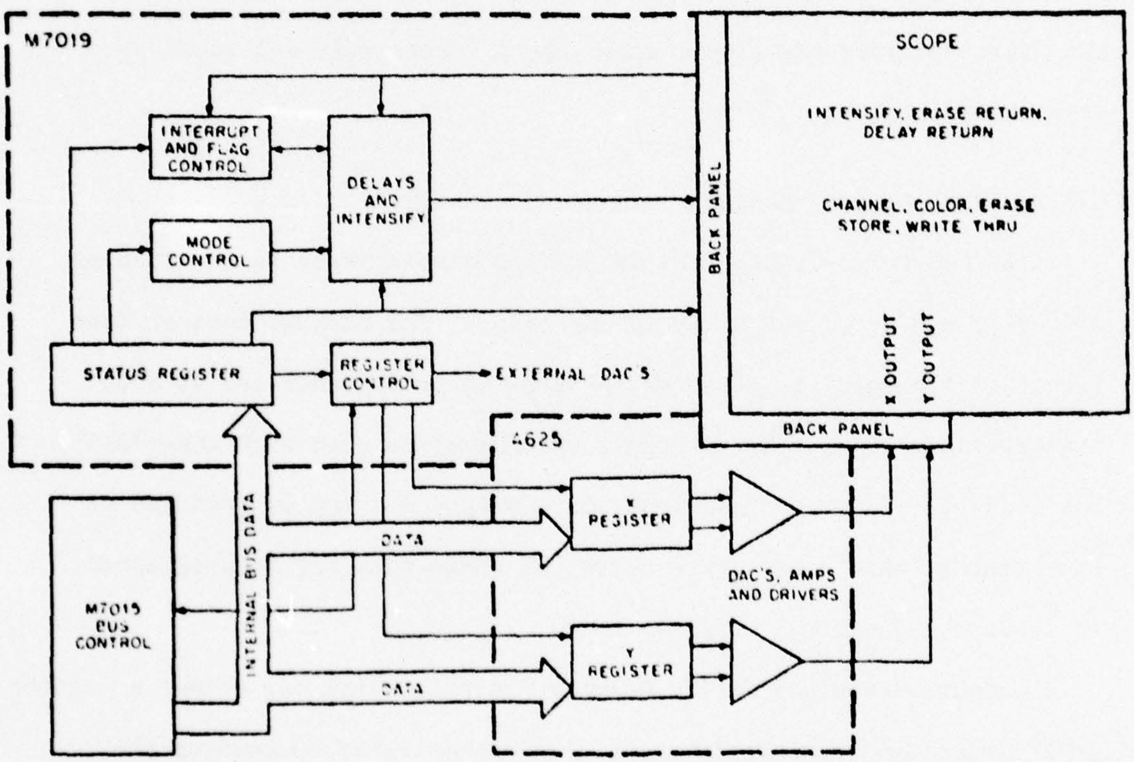
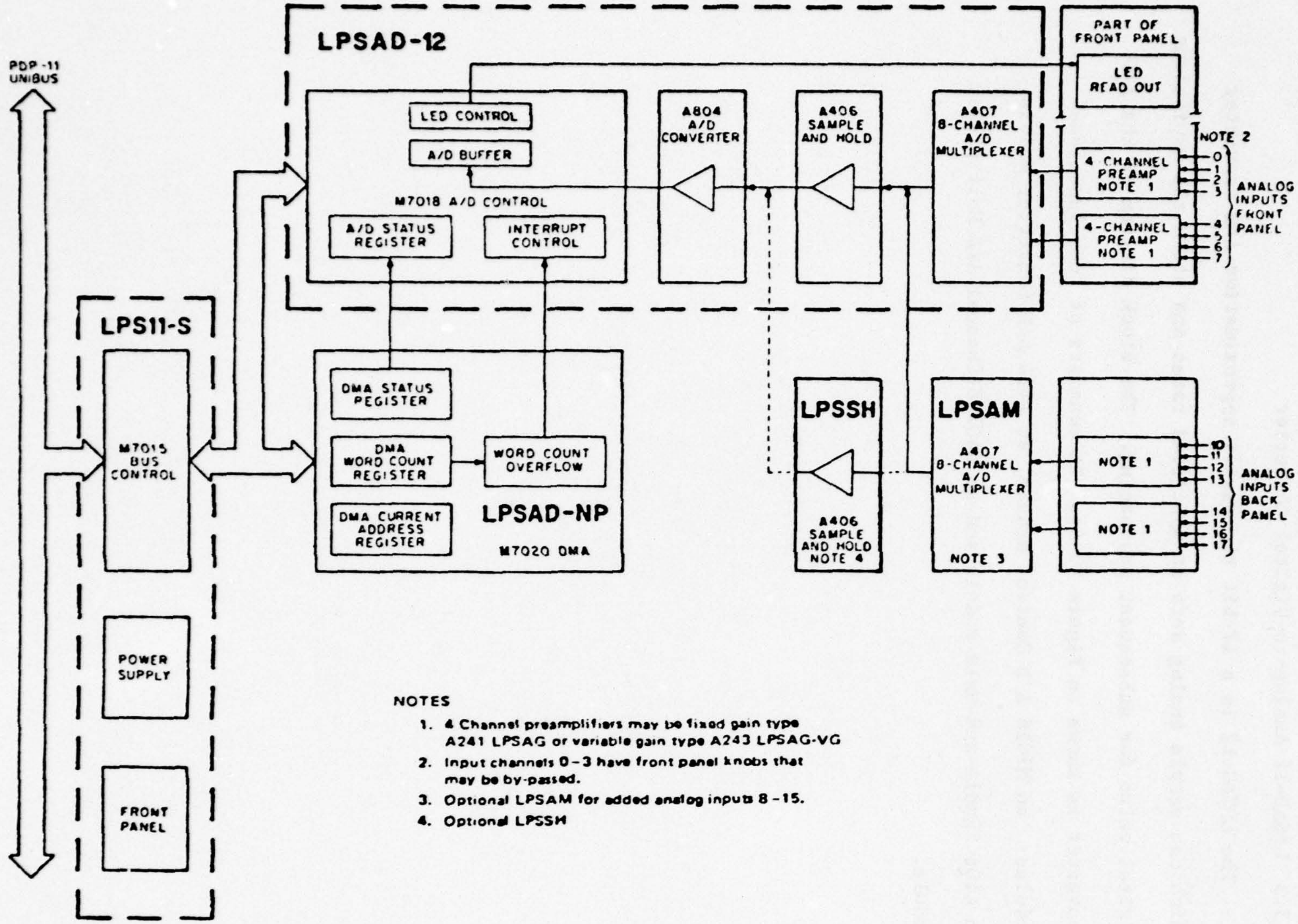


Figure (3-5) Display Control Block Diagram

3.3.3 LPSAD-12 Analog-to-Digital Converter

The LPSAD-12 is a 12-bit successive approximation A/D converter which can sample analog data at specified rates and store the equivalent digital value for subsequent processing. The block diagram of the A/D converter is shown in Figure (3-6). It consists of four functional modules: an M7018 A/D Control module, an A804 A/D Converter module, an A406 Sample-and-Hold module and an A407 8-Channel A/D Multiplexer module.



NOTES

1. 4 Channel preamplifiers may be fixed gain type A241 LPSAG or variable gain type A243 LPSAG-VG
2. Input channels 0 - 3 have front panel knobs that may be by-passed.
3. Optional LPSAM for added analog inputs 8 - 15.
4. Optional LPSSH

Figure (3-6) . A/D Converter Block Diagram

SECTION 4

DATA REDUCTION PROGRAMMING

This section discusses the implementation of the software programming for reducing the electrical conductivity data. All of the programs were written in Fortran IV and Assembly Language under the RT-11 system. Flow charts and listings of the programs are given in Appendices B and C, respectively.

4.1 Pulse Frequency and Timing Measurements

In determining the pulse frequency of the data waveform the Real-Time Clock is used to measure the time interval between the leading edges of consecutive pulses. The programming of the Real-Time Clock uses the status register and buffer/preset register, which are designated by 16-bit words and located at the addresses of 170404 and 170406 respectively. When the status register is loaded, it enables the counter to count at a designated rate; it controls the rate of the base frequency (100 Hz to 1 MHz); it causes an interrupt if its flag is set; and it counts the event timing from zero base by starting when a pulse comes from Schmitt trigger 2. Schmitt trigger 2 is used to shape the data waveform by converting the negative going data pulses to a train of negative pulses with the same frequency and a pulse width of approximately 2 μs. Each pulse from Schmitt trigger 2 sets the interrupt flag and causes the transfer of the contents of the buffer/preset register to a temporarily reserved

buffer in memory, until all the signals on magnetic tape have been digitized and stored in digital form on the disk. Therefore, the number of counts is proportional to the time interval of two consecutive pulses and inversely proportional to the pulse frequency. For example, if the 100 kHz clock frequency is used, the number of counts for a 100 pulses per sec (pps) waveform is 1,000 and for a 250 pps waveform, it is 400.

However, if the time interval of two consecutive pulses is greater than 0.65535 sec and the base frequency of the clock is 100 kHz, the number of counts will be greater than 65535_{10} which is the maximum value that the 16-bit buffer/preset register can handle. An interrupt service routine and an interrupt waiting loop are used when a time interval between two consecutive pulses is greater than 0.65535 sec. An index number within the interrupt waiting loop is set proportional to the number of times that the waiting loop has been completed between the two consecutive pulses. Also, the instruction time of the waiting loop can be calculated. Thus, an estimate of the waiting loop's total execution time actually gives the time interval of the two consecutive pulses. For example, if the instruction time for a complete waiting loop is 23.2±10% μs and the waiting loop has been completed $28,248_{10}$ times, the index number will decrease by 1. Thus, if the index number is -100, the time interval of these two pulses is 65.535 seconds.

The "WRITE" request is a "programmed request" that is an assembler macro call written into the program and interpreted by the PDP 11/10's monitor at the program execution time. It is used to

transfer a specific number of data points from a temporarily reserved buffer of memory to the disk. The control of the Central Processor Unit returns to the program immediately after the request is queued (<1 ms). The storage of data points requires double-buffered I/O techniques, i.e., the contents of one of the buffers are transferred from memory to the disk while the other buffer is filled immediately without interacting with the previous buffer. Since the data are digitized and collected from the tape recorder which is continuously running, the "WRITE" request interrupts the Central Processor Unit for less than 1 ms, which means that only pulses greater than 1 kHz will be lost.

Since each block of the DEC RK11/RK05 disk contains 400_8 (256_{10}) 16-bit words, the number of words of every buffer reserved in memory is usually an integer multiple of 256_{10}, i.e., every buffer could contain 14400_8 (6400_{10}) 16-bit words (25 blocks).

The "TTYIN" request is used to receive the characters from the LA36 DEC writer II. All of the characters received are in the form of ASCII Code. Thus, a subroutine "TTIN" is used to convert the characters from ASCII Code to numerical values. For example, the 16-bit word 34465_8 (the higher byte is 071_8 and lower byte 065_8) in ASCII Code converts to a numerical value of 59_{10}.

Typically, the data recorded on magnetic tape for one hour occupy about 900 blocks (230,400 16-bit words) on the disk, of which approximately 200 blocks are for the preflight calibration waveforms, 80 blocks for the data between the launch of the rocket and the separation of the payload from the rocket, and the remaining 620 blocks for the in-flight data.

4.2 Waveform Segmentation and Restorage

For an easy access to each waveform for data processing, the data are segmented on an individual waveform basis with the data points stored in seven blocks (per waveform). The restored data are assigned a different file name on the disk, and the original data file is used as a back-up.

For a typical experiment, usually ten to twenty preflight calibration waveforms and one hundred or more in-flight data waveforms are recorded on a magnetic tape. Referring to Figure (2-3), most waveforms contain three portions: a beginning portion of low frequencies, a region of increasing frequencies, and a final portion of relatively high and almost constant frequencies. Segmenting the data into individual waveforms was accomplished by recognizing the general features of the beginning portion of the next waveform. In doing this, several hundred to a thousand or more data points are extracted for a complete calibration or in-flight waveform.

4.3 Data Display

Although each data waveform has certain general characteristics, the large variability of the central portion of the waveform leads to no simple and dependable method of software programming to extract the linear regions which will provide the values of $(df_{\pm}/dt)_{DATA}$ (from which the electrical conductivity is derived). An example of how a complete waveform is displayed is shown in Figure (4-1). The values of the horizontal coordinate indicate the sequential number of the data points and the scale is a function of time. The values and scale of

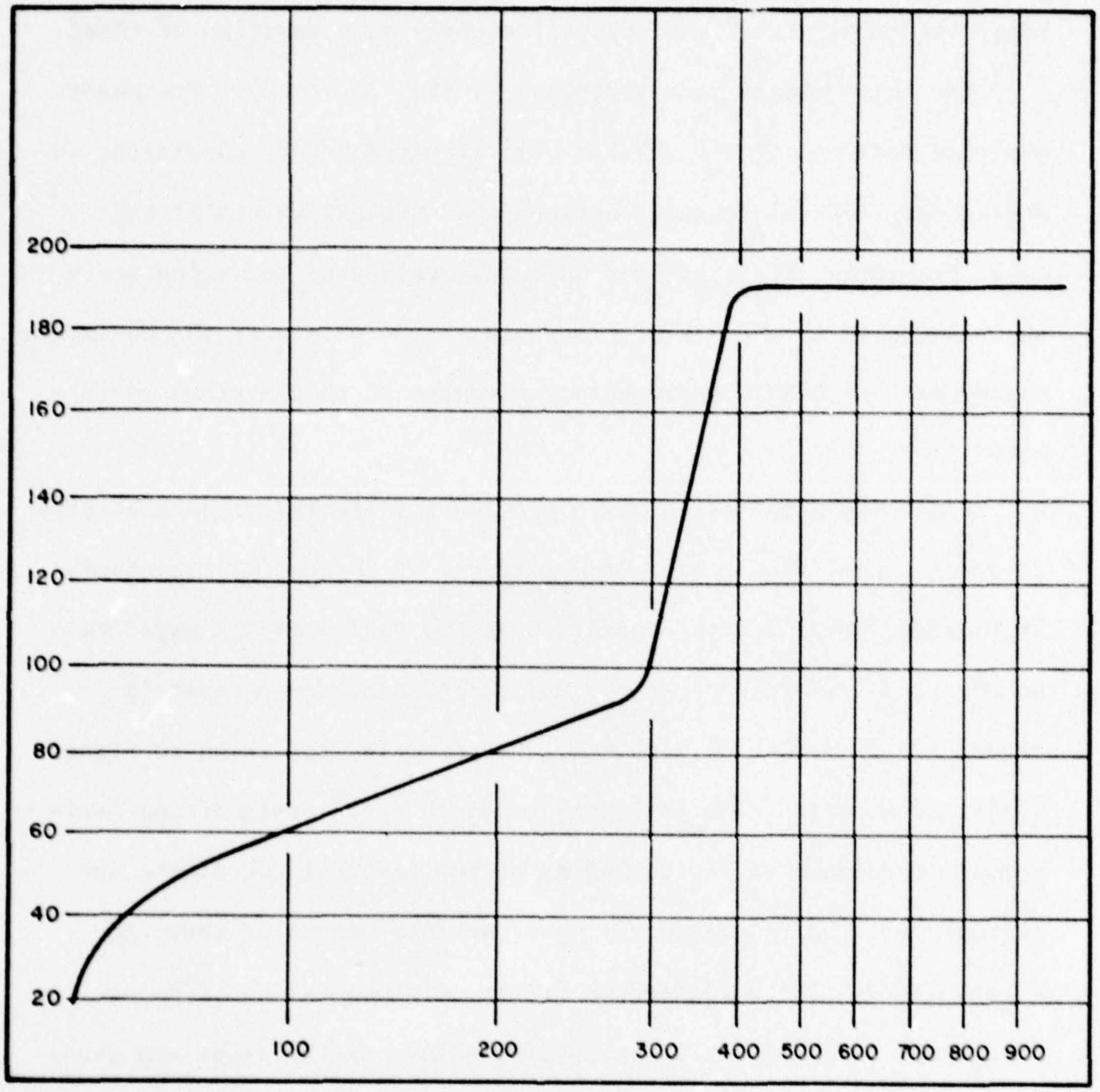


Figure (4-1) Display of a Complete Waveform on the Scope

the vertical coordinate are a function of frequency in Hz. Therefore, the scope actually displays frequency as a function of time.

The scope matrix is a $4,096_{10} \times 4,096_{10}$ dot array. Frequency and time scalings of the waveform are required before displaying on the scope. The relationship between the original values of the data, frequency, time, and corresponding values on the scope are shown in Table 1, where LPSVCX represents the values of the horizontal coordinate and LPSVCY represents the values of the vertical coordinate.

Since the data points which provide for electrical conductivity information are within a limited interval of the overall waveform, it is often desirable to expand a specific region of the waveform on the scope in order to better understand and more accurately choose the important data points. For example, referring to Figure (4-1), if a region from the 200th to 400th data points of the horizontal coordinate and 60 to 100 Hz of the vertical coordinate are redisplayed, the resulting expanded waveform occurs as shown in Figure (4-2). The relationship of the original values of data, frequency, time and the corresponding values on the scope are provided in Table 2. Identification of a waveform segment requires numbering all of the data points in sequential order. The numerical characters (from 0 to 9) are also displayed on the scope and are constructed using discrete dots on a 5 x 7 grid as shown in Figure (4-3).

TABLE 1: Original Scalings on the Scope

Sequential Number of Data Points	Value of Data (Number of Count)	Frequency (Hz)	Time (ms)	LPSVCX	LPSVCY
1	5,000	20	50.00	1250	320
2	2,500	40	75.00	1875	640
3	1,666	60	91.66	2291	960
4	1,250	80	104.16	2603	1280
5	1,000	100	114.16	2853	1600
6	833	120	122.49	3061	1920
7	714	140	129.63	3239	2240
8	625	160	135.88	3395	2560
9	555	180	141.43	3533	2880
10	500	200	146.43	3658	3200

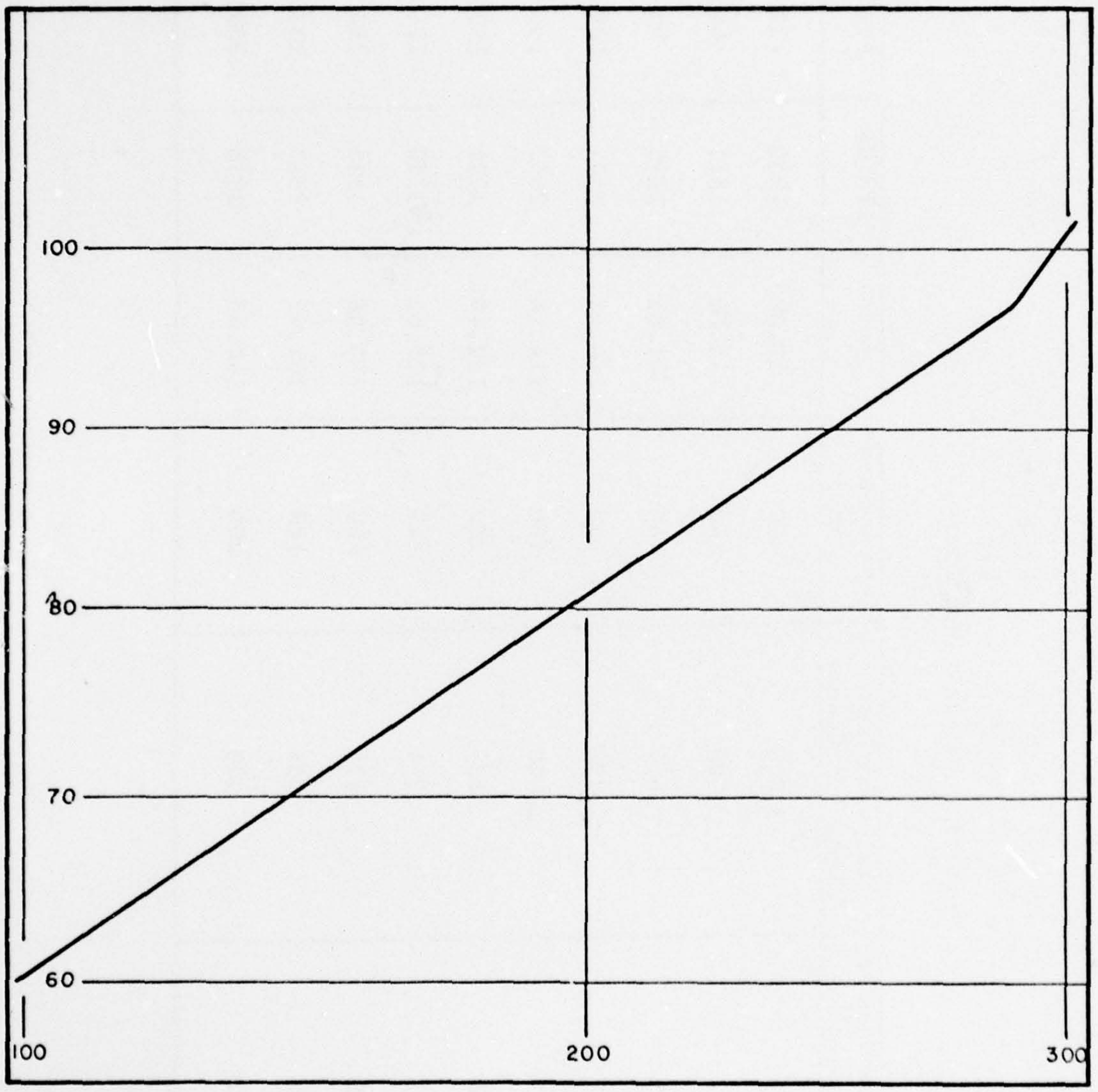


Figure (4-2) Display of a Specific Region of a Waveform on the Scope

TABLE 2: Expanded Scalings on the Scope

Sequential Number of Data Points	Value of Data (Number of Count)	Frequency (Hz)	Time (ms)	LPSVCX	LPSVCY
1	5,000	20	-	-	-
2	2,500	40	-	-	-
3	1,666	60	16.66	1666	400
4	1,250	80	29.16	2916	1800
5	1,000	100	39.16	3916	3200
6	833	120	-	-	-
7	714	140	-	-	-
8	625	160	-	-	-
9	555	180	-	-	-
10	500	200	-	-	-

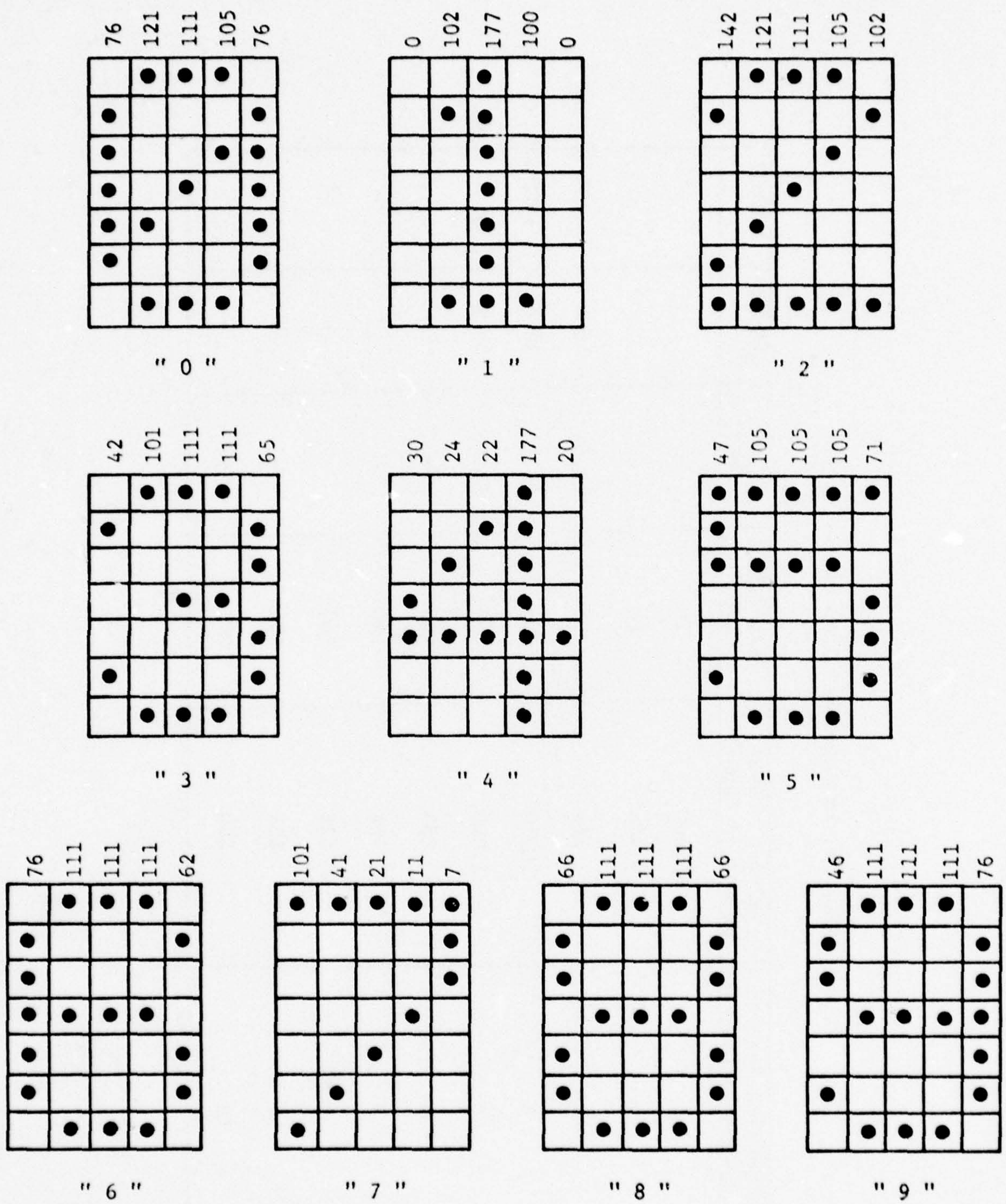


Figure (4-3) Display of Numerical Characters

4.4 Data Processing

According to Eqs. (2-5) and (2-6), the electrical conductivity information is obtained from the values of $(df/dt)_{CAL}$ and $(df_{\pm}/dt)_{DATA}$ which are the slopes of specific regions of the calibration and data waveforms, respectively. The slopes are obtained by fitting a one-independent variable equation, i.e., a straight line, to the designated data points of the waveform. The equation is expressed as follows:

$$f = A\,t + B \tag{4-1}$$

where f is the frequency in Hz, t is the time in seconds, A is the slope of the straight line in Hz/second (which is also the value of (df/dt)) and B is the intercept in Hz. The straight line generated by this equation is also displayed on the scope to assure that it fits within the specified tolerance. Special techniques for determining a least-squares straight line fit to the data are presented in Section 5.

4.5 Timing Data

The related real time (referenced to the payload's launch) at which a data waveform is scaled can be obtained by accumulating the interval for all of the data pulses. This time information is important in later determining the payload's altitude from radar data, which are also usually referenced to launch time.

There are three possible cases to consider in converting the number of counts of data to the value of real time. If the number

of counts of a data point is less than $32,767_{10}$ (077777_8), the real time (T_r) for this data point is simply accumulated as follows:

$$T_r = \frac{N}{100,000} + T_b \tag{4-2}$$

where T_b is the real time accumulated before this data point, N is the number of the counts of this data point. If the number of counts is greater than $32,767_{10}$ (077777_8) and less than $65,535_{10}$ (177777_8), the real time for this data point has to be expressed as:

$$T_r = \frac{(65,536 + N)}{100,000} + T_b \tag{4-3}$$

If the number of counts is greater than $65,535_{10}$ (177777_8), the data acquisition system generates a particular value (-100_{10}), which is stored before the index number in order to be easily recognized and the index number "I" is transferred to the real time as follows:

$$T_r = 0.65535_{10} \times (-I) + T_b \tag{4-4}$$

where the index number I is stored in the data acquisition system as a negative value. The time determined by accumulating the data counts is estimated to be within 1% of the actual time value, which is considered satisfactory for a subsonic experiment.

SECTION 5

CURVE FITTING TECHNIQUES

This section discusses a method of fitting a straight line equation to a mass of data [Daniel and Wood (1971)]. The method for determining a straight line fit should use all of the relevant data in estimating each constant of the equation, have reasonable economy in the number of constants required, provide some estimate of the uncontrolled error and give some idea of how well the final equation can be expected to predict the response.

5.1 Linear Least-Squares Method

The most popular method of fitting an equation to a mass of data is the least-squares method. This method finds the values of the constants in the designated equation such that the sum of the squared deviations of the observed values from those predicted by the equation is minimized.

From the data pairs (t_{n1}, f_{n1}), (t_{n1+1}, f_{n1+1}), ...,(t_{n2}, f_{n2}) of the specified region of the waveform (see Figure (5-1)), there are four assumptions about the relationship between the observed value of the independent variable t_i and the observed value of the dependent variable f_i for determining a straight line $y = A\,x + B$ to fit the data:

1. There is a linear relationship between the predicted value of a response y and the value of the independent variable x

$$y = A\,x + B \qquad (5\text{-}1)$$

where A is the slope and B is the intercept.

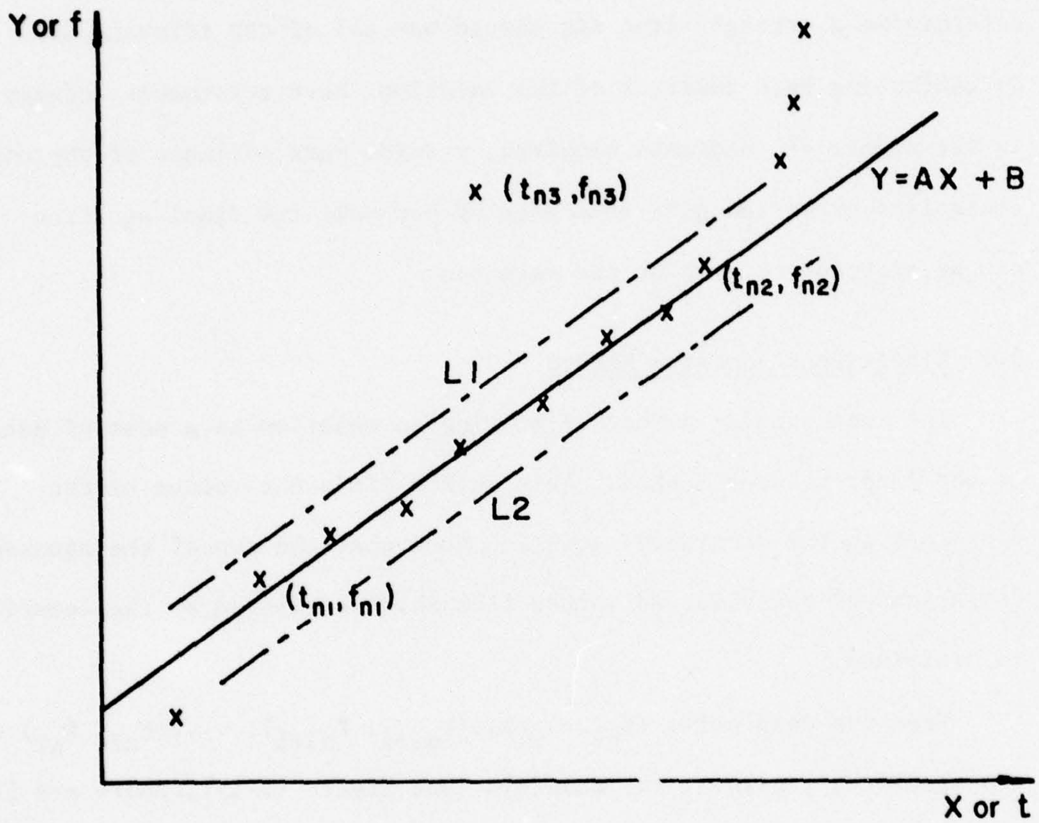


Figure (5-1) Confidence Region of Least-Squares Estimates

2. The observed value $f_i = y_i + e_i = A\,x_i + B + e_i$, where e_i is the random error.

3. The random error e_i has the following properties:

   1) The expected value of e_i is zero and the observed f_i is an unbiased estimate of y_i.

   2) The variance of e_i remains constant for all values of x_i.

   3) The e_i values are statistically uncorrelated.

4. The observed values of the independent variable are measured without error. All of the error is thus in f_i and none is in the t_i's.

Therefore the linear least-squares estimates for a straight line are those values of A and B which minimize the function as follows:

$$Q = \sum_{i=n1}^{n2} (e_i)^2 = \sum_{i=n1}^{n2} (f_i - y_i)^2 = \sum_{i=n1}^{n2} (f_i - Ax_i - B)^2 \qquad (5\text{-}2)$$

From the above assumptions, there is no error for the independent variable t_i and hence, $x_i = t_i$. Thus we obtain

$$Q = \sum_{i=n1}^{n2} (f_i - At_i - B)^2 \qquad (5\text{-}3)$$

Upon setting

$$\frac{\partial Q}{\partial A} = 2 \sum_{i=n1}^{n2} (f_i - At_i - B)\,(-t_i) = 0 \qquad (5\text{-}4)$$

and

$$\frac{\partial Q}{\partial B} = 2 \sum_{i=n1}^{n2} (f_i - At_i - B)(-1) = 0 \tag{5-5}$$

then

$$B \sum_{i=n1}^{n2} t_i + A \sum_{i=n1}^{n2} t_i^2 = \sum_{i=n1}^{n2} t_i f_i \tag{5-6}$$

and

$$BN + A \sum_{i=n1}^{n2} t_i = \sum_{i=n1}^{n2} f_i \tag{5-7}$$

where $N = (n2 - n1 - 1)$, which is the total number of data points in the above equation.

Hence

$$A = \frac{N \sum_{i=n1}^{n2} t_i f_i - \sum_{i=n1}^{n2} t_i \sum_{i=n1}^{n2} f_i}{N \sum_{i=n1}^{n2} t_i^2 - \sum_{i=n1}^{n2} t_i \sum_{i=n1}^{n2} f_i} \tag{5-8}$$

and

$$B = \frac{\sum_{i=n1}^{n2} f_i \sum_{i=n1}^{n2} t_i^2 - \sum_{i=n1}^{n2} t_i f_i \sum_{i=n1}^{n2} t_i}{N \sum_{i=n1}^{n2} t_i^2 - \left(\sum_{i=n1}^{n2} t_i \right)^2} \tag{5-9}$$

Thus, the predicted value y_i is a function of the known data pairs (t_i, f_i) only and can be expressed as follows:

$$y_i = A\,x_i + B = A\,t_i + B$$

$$= \frac{N \sum_{i=n1}^{n2} t_i f_i - \sum_{i=n1}^{n2} t_i \sum_{i=n1}^{n2} f_i}{N \sum_{i=n1}^{n2} t_i^2 - \sum_{i=n1}^{n2} t_i \sum_{i=n1}^{n2} f_i}\, t_i$$

$$+ \frac{\sum_{i=n1}^{n2} f_i \sum_{i=n1}^{n2} t_i^2 - \sum_{i=n1}^{n2} t_i f_i \sum_{i=n1}^{n2} t_i}{N \sum_{i=n1}^{n2} t_i^2 - \left(\sum_{i=n1}^{n2} t_i\right)^2} \tag{5-10}$$

Finally the straight line $y = A\,x + B$ is derived in which the slope A is the value of $(df/dt)_{CAL}$, $(df_+/dt)_{DATA}$ or $(df_-/dt)_{DATA}$ in Eq. (2-5) or (2-6).

5.2 Confidence Region

The digital data acquired from magnetic tape in general contain some unexpected noisy data. Therefore a confidence region is defined by an appropriate factor which corresponds to the error between the observed value f_i and the predicted value y_i of the fitted equation. According to Figure (5-1), the confidence region is between the two straight lines L1 and L2 for which every point can be expressed as

follows:

$$L1: \quad Y_{L1} = y_i + e_o \tag{5-11}$$

$$L2: \quad y_{L2} = y_i - e_o \tag{5-12}$$

where e_o is the value chosen to define the confidence region. For the jth data point within the specified region of the waveform, the error e_j between the predicted and observed values is

$$e_j = \left| y_i - f_j \right| \tag{5-13}$$

If $e_j > e_o$, obviously the jth data point (such as (t_{n3}, f_{n3}) in Figure (5-1)) is classified as a noisy data point and is not included in Eqs. (5-2) through (5-10). Therefore, the error caused by the unexpected noisy data point is eliminated.

5.3 Residual Root Mean Square

The residual root mean square is computed to determine how well the final equation can be expected to predict the response. The value of the residual root mean square is expressed as

$$\text{RRMS} = \left(\frac{1}{N_o} \sum_{\substack{i=n1 \\ i \neq j1, j2, \ldots}}^{n2} (y_i - f_i)^2 \right)^{1/2} \tag{5-14}$$

where the j1th, j2th,... data points are not included in Eqs. (5-2) through (5-10) to obtain A and B of Eq. (5-1). The final number of total data points N_o of Eqs. (5-8) and (5-9) is always less than or equal to N.

SECTION 6

DISCUSSION AND CONCLUSIONS

6.1 Electrical Conductivity Measurements

The computerized reduction system, as described in the previous sections, was used to reduce blunt probe electrical conductivity data obtained from a rocket experiment launched at 1230 MST on September 28, 1976 from White Sands Missile Range, New Mexico. This particular rocket experiment was conducted in conjunction with the STRATCOM VIIA balloon flight launched from Holloman Air Force Base, New Mexico. The dots in Figure (6-1) represent the conductivity values obtained using the reduction program. Smooth curves have been fitted to the data.

Above 40 km, the profiles diverge with the negative conductivity values larger than the corresponding positive conductivity measurements. In this region, the negative charge particles are generally more mobile than the positive ions. Below 40 km, the positive and negative conductivity values for the same altitude are comparable.

The pulses in Figure (6-1) represent conductivity values resulting from manually scaling the demodulated waveforms (see Figure (2-3)) to obtain the values of $(df_{\pm}/dt)_{DATA}$ and $(df/dt)_{CAL}$. This is the procedure formerly used for reducing electrical conductivity data [Hale and Hoult (1965); Mitchell (1973)].

It is also important to note that the computer reduction method has expanded the altitude region over which the conductivity data waveforms can be reduced. At higher altitudes where the slopes of the data waveforms are the largest, it is very difficult to manually scale them

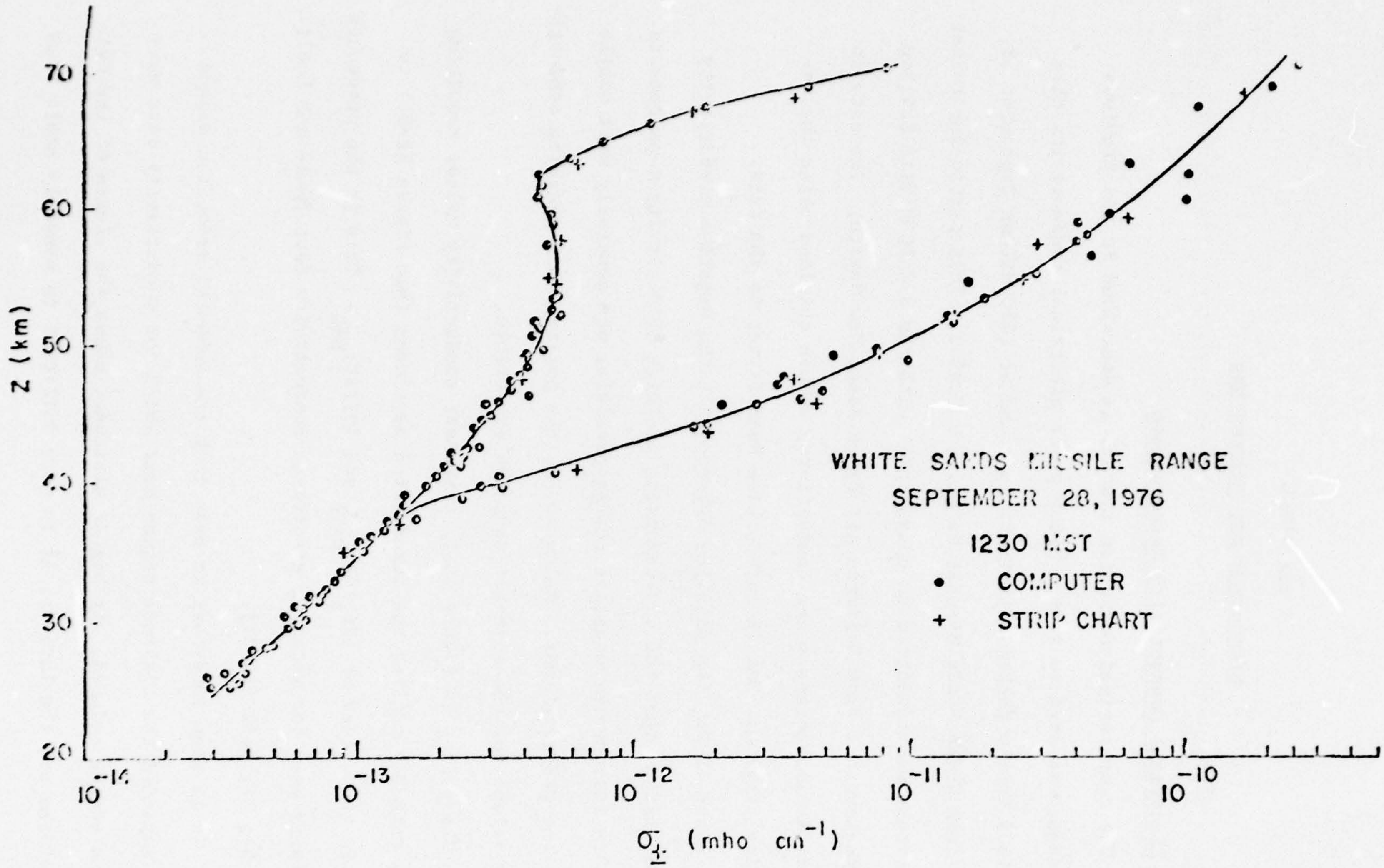


Figure (6-1) Parachute-Borne Blunt Probe Electrical Conductivity Data

from the strip chart. Also, the response times of the data demodulation system and the strip chart recorder can introduce possible errors. At lower altitudes where the data waveforms' slopes are very small, it is also very difficult to manually scale the data. By using the computerized scheme to reduce the data for the September 28, 1976, blunt probe experiment, values were obtained for the electrical conductivity in the 25 to 70 km altitude region where this parameter is observed to change by approximately four orders of magnitude.

Another check of this reduction scheme is to compare the results of the rocket flight with the data from the STRATCOM VIIA blunt probe experiment, which are shown in Figure (6-2). Included in this figure are the smooth curves for the rocket data in Figure (6-1) and the balloon blunt probe values obtained from 1100 to 1800 MST on September 28, 1976, while the balloon slowly descended from 39 km to 19 km. The balloon data format made it more suitable for reduction by manually scaling the waveforms from a strip chart. Again, very good agreement was observed between the two sets of data. This hopefully suggests consistency between both the data reduction procedures and the experiment techniques.

6.2 Conclusions

A computerized system using the DEC PDP 11/10 minicomputer and associated peripherals has been developed for reducing subsonic blunt probe and Gerdien condenser electrical conductivity data. Assembly Language and FORTRAN IV programs were written under the DEC RT-11 operating system to perform data digitizing, acquisition, storage,

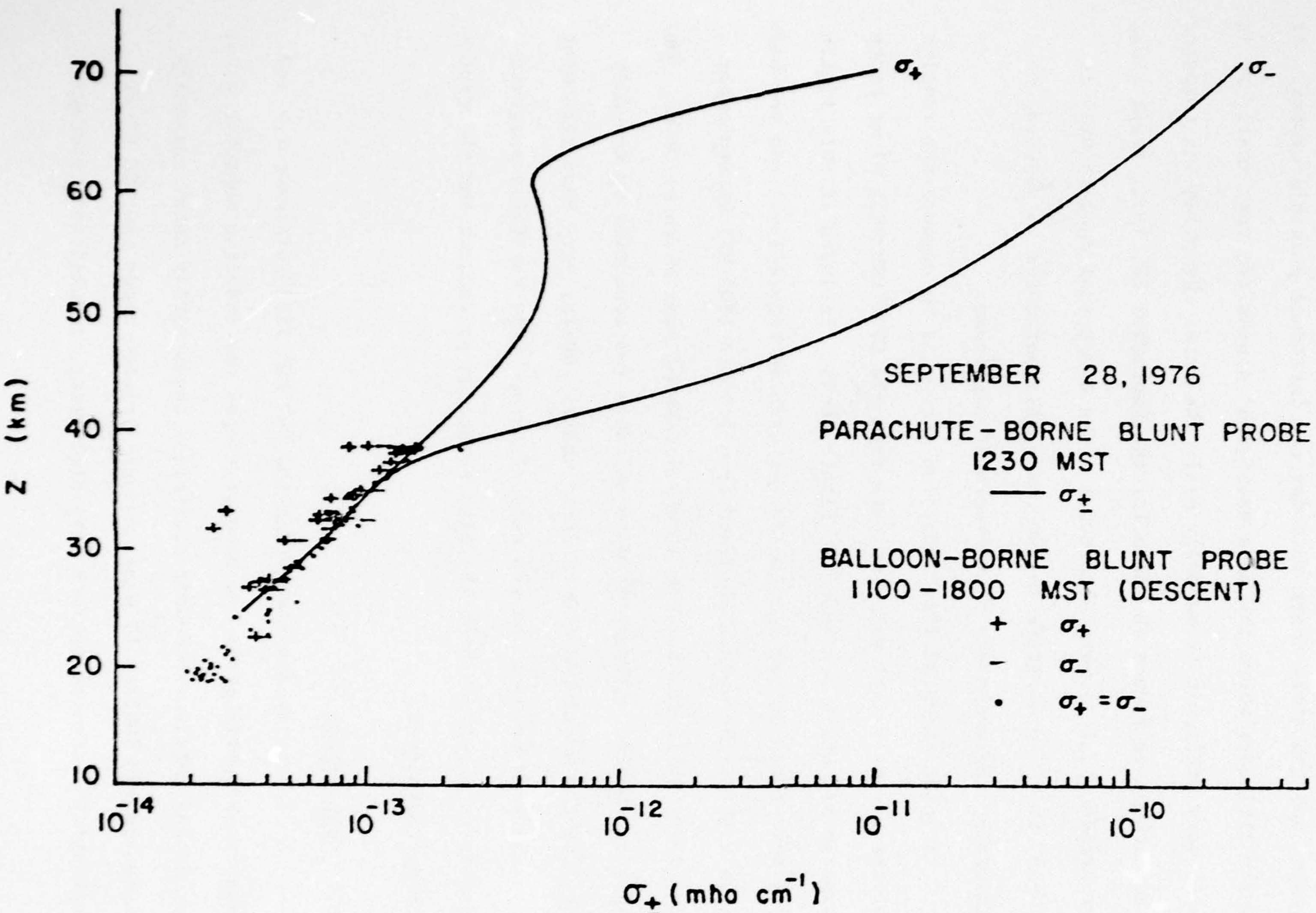


Figure (6-2) Balloon and Parachute-Borne Blunt Probe Electrical Conductivity Data

display, processing and printing out the results. Interaction with the system is necessary for interpreting the data waveforms and for choosing the particular segments of the waveforms to be scaled.

The electrical conductivity values determined by using this technique on rocket blunt probe data were found to be consistent with the values obtained by manually scaling the data waveforms. In addition, the enhanced accuracy of the computerized system makes it possible to scale the data over a larger altitude region, i.e., a wider range of values, than was found possible using the manual technique.

In the future, a noninteractive system using a microprocessor with a minicomputer on-line to handle the data acquisition and processing simultaneously would provide a substantial savings in the time required for data reduction. Implementation of such a system, however, would require considerable attention to developing the computer software for interpreting the probe's current-voltage response.

REFERENCES

Baker, D.C., and L.C. Hale, D-region parameters from blunt probe measurements during a solar eclipse, Space Res. X, North-Holland, Amsterdam, 712, 1970.

Conley, T.D., Mesospheric positive ion concentration mobility and loss rates obtained from rocket-borne Gerdien condenser measurements, Radio Sci. 9, 575, 1974.

Croskey, C.L., In situ measurements of the mesosphere and stratosphere, Scientific Report No. 442, Ionosphere Research Laboratory, The Pennsylvania State University, 1976.

Croskey, C.L., L.C. Hale, and S.F. Lieden, Results of ionization measurements in the middle atmosphere, Space Res. XVII, in press, 1977.

Daniel, C.T., and F.S. Wood, Fitting Equations to Data, Wiley-Interscience, New York, 1971.

Farrokh, H., Design of a simple Gerdien condenser for ionospheric D-region charged particle density and mobility measurements, Scientific Report No. 433, Ionosphere Research Laboratory, The Pennsylvania State University, 1975.

Hale, L.C., Parameters of the low ionosphere at night deduced from parachute borne blunt probe measurements, Space Res. VII, North-Holland, Amsterdam, 140, 1967.

Hale, L.C., Positive ions in the mesosphere, Space Res. XIV, Akademie-Verlag, Berlin, 219, 1974.

Hale, L.C., and D.P. Hoult, A subsonic D-region probe theory and instrumentation, Scientific Report No. 247, Ionosphere Research Laboratory, The Pennsylvania State University, 1965.

Hale, L.C., D.P. Hoult, and D.C. Baker, A summary of blunt probe theory and experimental results, Space Res. VIII, North - Holland, Amsterdam, 320, 1968.

Hoult, D.P., D-region probe theory, J. Geophys. Res. 70, 3183, 1965.

LPS11 Laboratory Peripheral System User's Guide, Digital Equipment Corporation, Maynard, Massachusetts, 1973.

Mitchell, J.D., An experimental investigation of mesospheric ionization, Scientific Report No.416, Ionosphere Research Laboratory, The Pennsylvania State University, 1973.

Mitchell, J.D., and L.C. Hale, Observations of the lowest ionsphere, Space Res. XIII, Akademie-Verlag, Berlin, 471, 1973.

Mitchell, J.D., L.C. Hale, and C.L. Croskey, Electrical conductivity measurements in the stratosphere using balloon and parachute-borne blunt probes, To be presented at the XXth COSPAR meeting, Tel-Aviv, 1977.

Mitchell, J.D., L.C. Hale, R.O. Olsen, J. Randhawa, and R. Rubio, Positive ions and the winter anomaly, Radio Sci. 7, 175, 1972.

Mitchell, J.D., R.S.Sagar, and R.O. Olsen, Positive ions in the middle atmosphere during sunrise conditions, Space Res. XVII, 199, 1977.

Olsen, R.O., Private Communication, 1977.

Olsen, R. O., J.D. Mitchell, and C.L. Croskey, Temperature and electrical conductivity measurements at a high latitude site during a period of auroral activity, EOS Trans. 57, 301, 1976.

PDP 11-04/05/10/35/40/45 Processor Handbook, Digital Equipment Corporation, Maynard, Massachusetts, 1975

PDP 11 Peripherals Handbook, Digital Equipment Corporation, Maynard, Massachusetts, 1975.

Pedersen, A., Measurements of ion concentrations in the D-region of the ionsphere with a Gerdien condenser rocket probe, FOA 3 Report A607, Research Institute of National Defense, Stockholm, Sweden, 1964.

Rose, G., and H.U. Widdel, Results of concentration and mobility measurements for positively and negatively charged particles taken between 85 and 22 km in sounding rocket experiments, Radio Sci, 7, 81, 1972.

RT-11 System Reference Manual, Digital Equipment Corporation, Maynard, Massachusetts, 1975.

Sagar, R.S., A subsonic Gerdien condenser experiment for upper atmosphere research, Master's thesis, The University of Texas at El Paso, El Paso, Texas, 1976.

The Meteorological Rocket Network Document 11-64, White Sands Missile Range, New Mexico, 1965.

Zimmerman, L.E., Integrated circuit electrometer and sweep circuitry for an atmospheric probe, Scientific Report No. 376(E), Ionosphere Research Laboratory, The Pennsylvania State University, 1971.

APPENDIX A

USER'S MANUAL

This section was written to assist the user in interfacing the system, using the software programs and obtaining the results.

A1. Data Acquisition System

1) Equipment

1.1 DEC PDP 11/10 Minicomputer

1.2 Real-Time Clock of LPS11 Laboratory Peripheral System

1.3 HP3960 Instrumentation Recorder (or equivalent)

1.4 LA36 DEC writer II

1.5 Dual trace Oscilloscope

2) Instructions

2.1 Connect the output terminal of the data channel of the HP3960 Instrumentation Recorder to both of the input terminals of Schmitt trigger 2 of the LPS11 Real-Time Clock and channel 1 of the dual trace oscilloscope.

2.2 Connect the output terminal of Schmitt trigger 2 of the LPS11 Real-Time Clock to the input terminal of channel 2 of the dual trace oscilloscope.

2.3 Set "-Slope" of Schmitt trigger 2.

2.4 Adjust the level of Schmitt trigger 2 until it fires in the correct position as observed on the oscilloscope.

3) Software Program Usage

3.1 Input "R NEW102" to the console of the LA36 DEC writer II.

3.2 Input the value of the number of blocks for every buffer to the console.

3.3 Input the values of the times of transferring data from memory to the disk to the console.

A2. Waveform Segmentation and Restorage

1) Equipment

1.1 DEC PDP 11/10 Minicomputer

1.2 Display Control of LPS11 Laboratory Peripheral System

1.3 Tektronix 603 Storage Scope

1.4 LA36 DEC writer II

2) Software Program Usage

2.1 Input "R NEW500" to the console of the LA36 DEC writer II.

2.2 Find the data at the launching of the rocket. Input "3" to the console.

2.3 Find the data between the launching and the separation of the payload from the rocket. Input "4" to the console.

2.4 Find the data at the separation of the payload from the rocket. Input "5" to the console.

2.5 Input the values of the number of blocks for every buffer, the total number of blocks of the input data file and the total number of segmented waveforms, respectively, to the console.

A3. Data Processing

1) Equipment

1.1 DEC PDP 11/10 Minicomputer

1.2 Display Control of LPS11 Laboratory Peripheral System

1.3 Tektronix 603 Storage Scope

1.4 LA36 DEC writer II

2) Software Program Usage

2.1 Input "R NEW900" to the console of the LA36 DEC writer II.

2.2 Input the sequential number of the first waveform to be processed to the console.

2.3 Input "1" to the console for expansion of the specific region of the waveform.

2.4 Input "2" to the console for obtaining the positive electrical conductivity.

2.5 Input "3" to the console for obtaining the negative electrical conductivity.

2.6 Input "4" to the console for the processing of the other waveform.

2.7 Input "5" for terminating the data processing and printing out the results.

2.8 Input the sequential numbers of the first and last data points and the tolerance of the frequency to the console for obtaining the electrical conductivity.

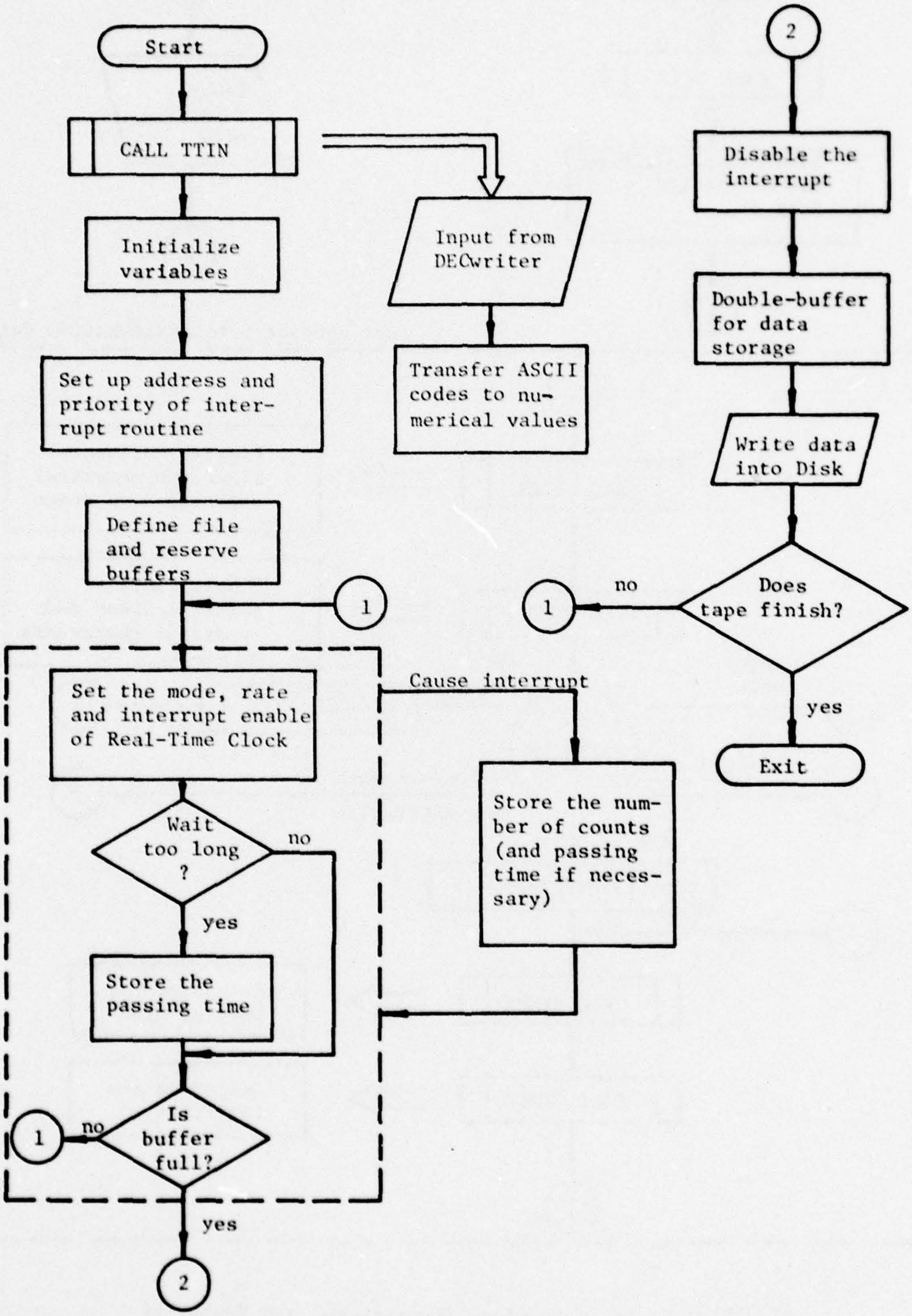
APPENDIX B

FLOW CHARTS OF MINICOMPUTER (RT-11) PROGRAMS

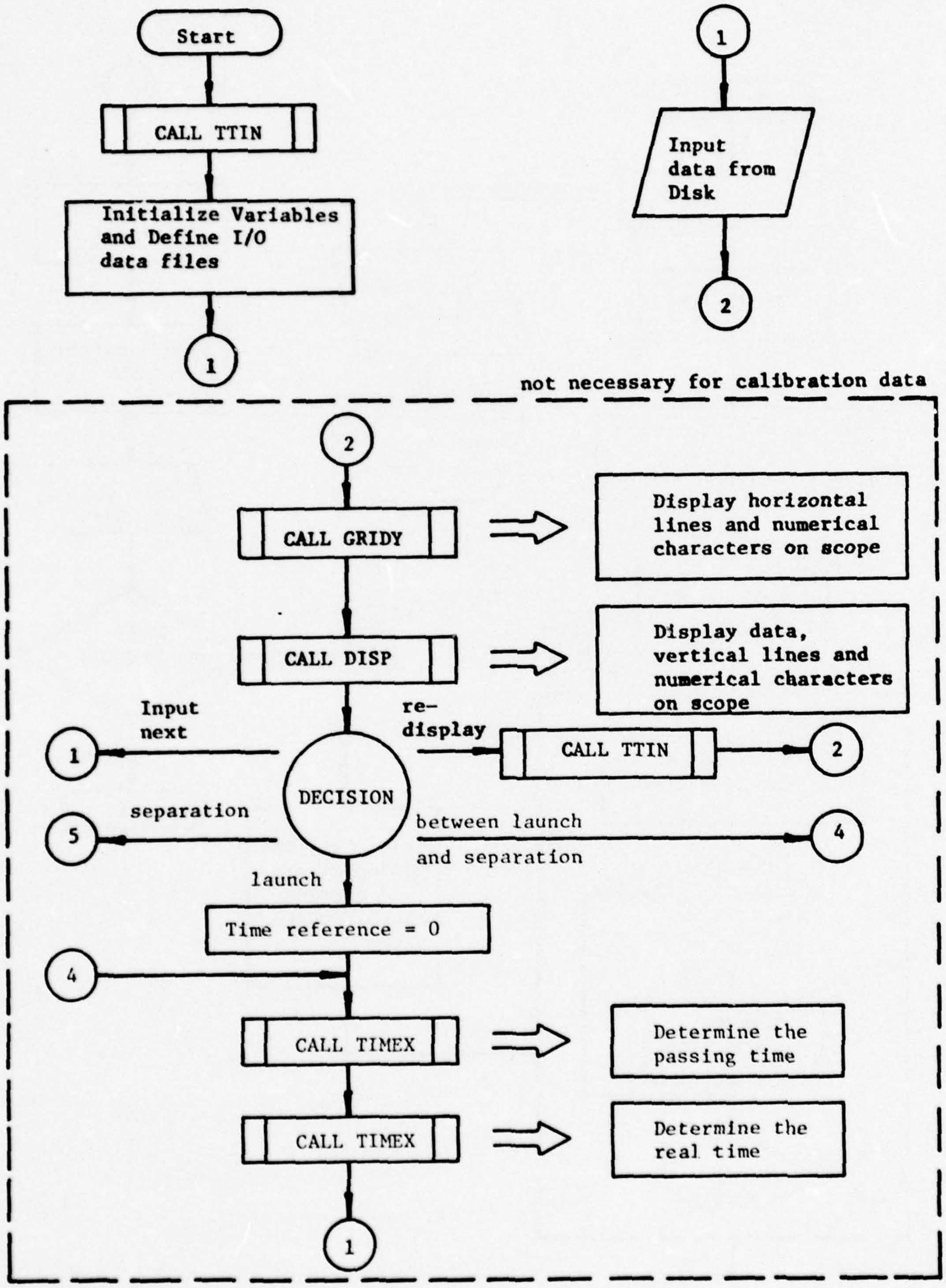
B1 - Flow chart of data acquisition.

B2 - Flow chart of waveform segmentation and restorage.

B3 - Flow chart of data processing (using a least-squares method) and printing out the results.

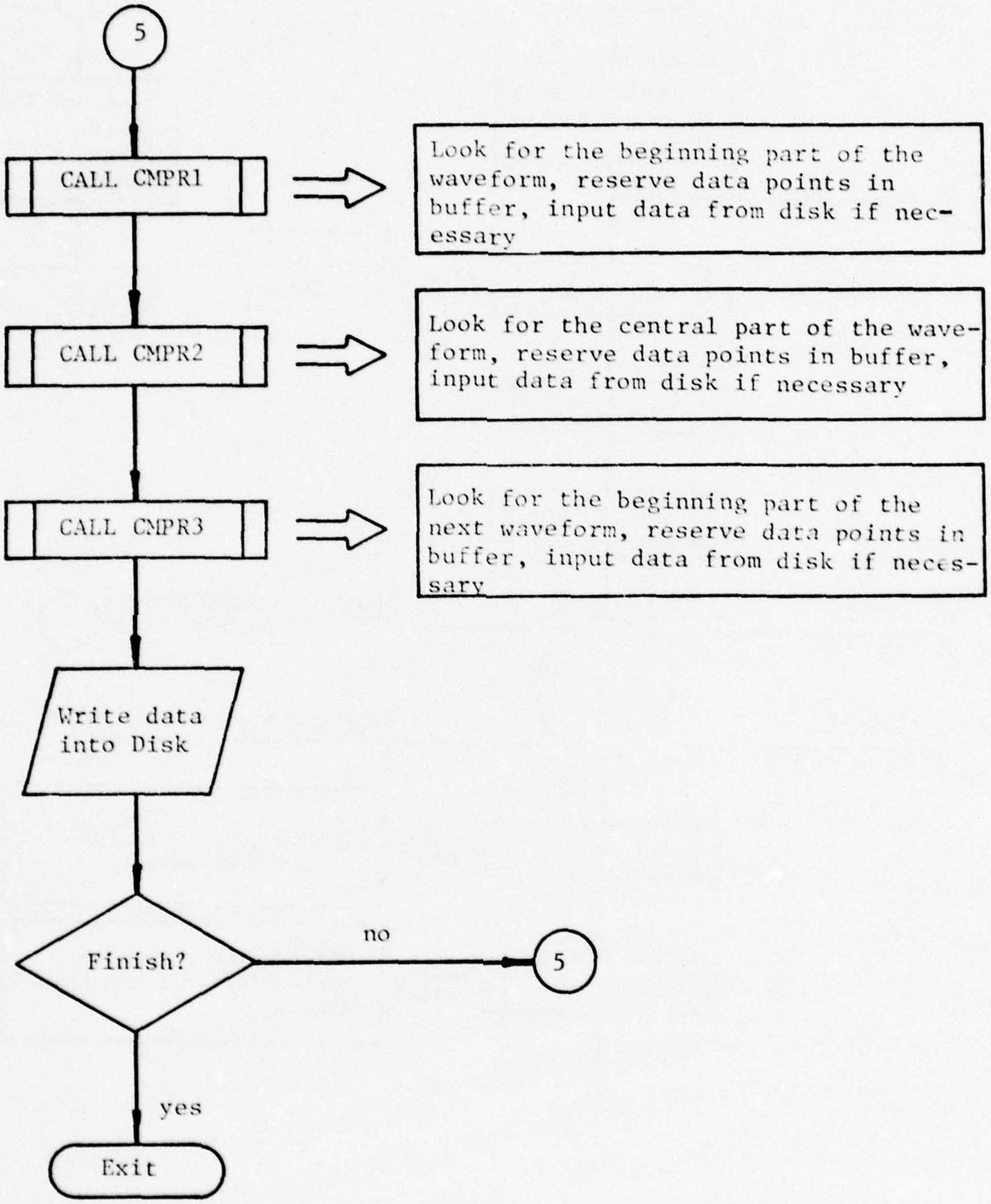


B1. Flow Chart of Data Acquisition



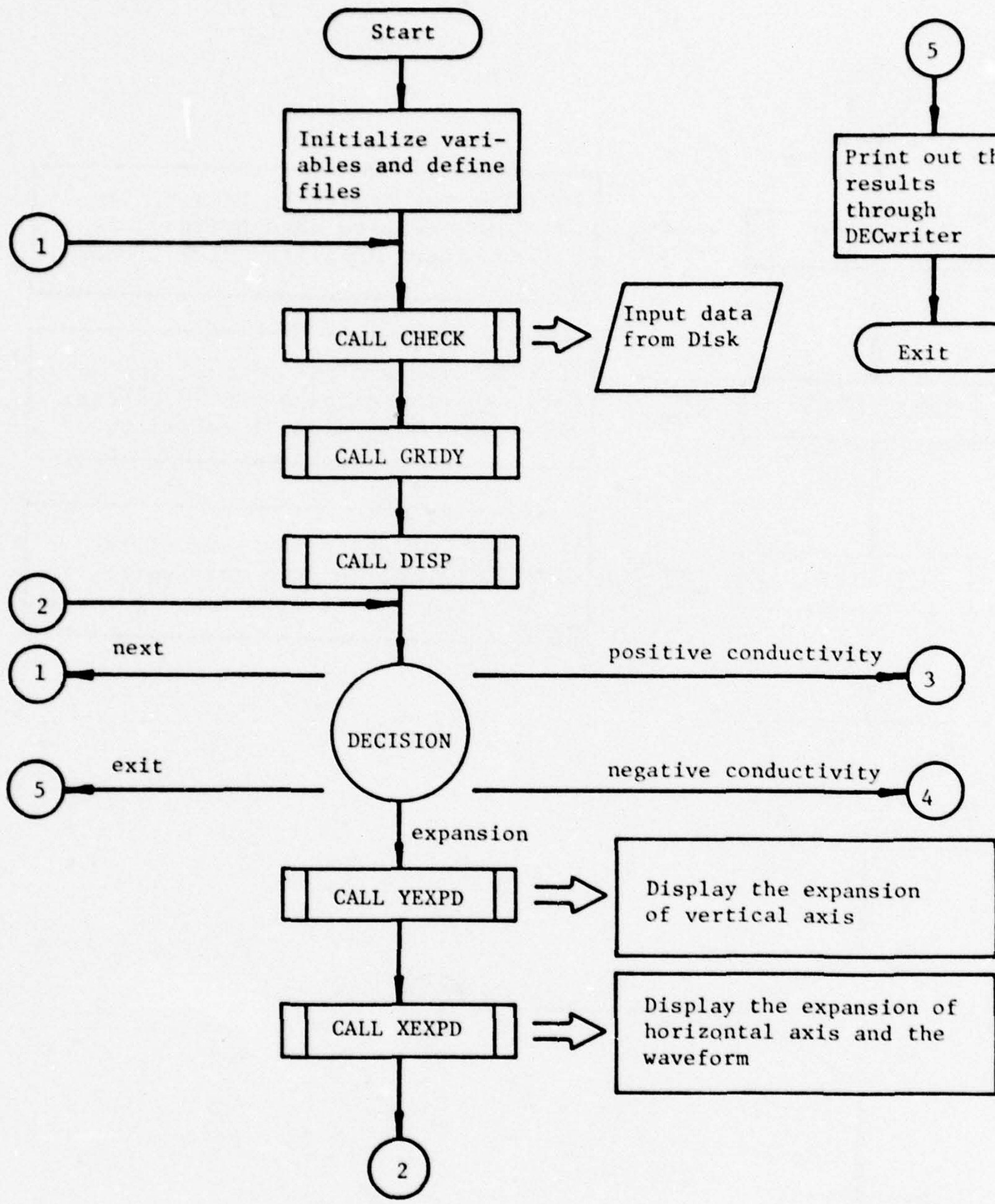
B2. Flow Chart of Waveform Segmentation and Restorage

APPENDIX B



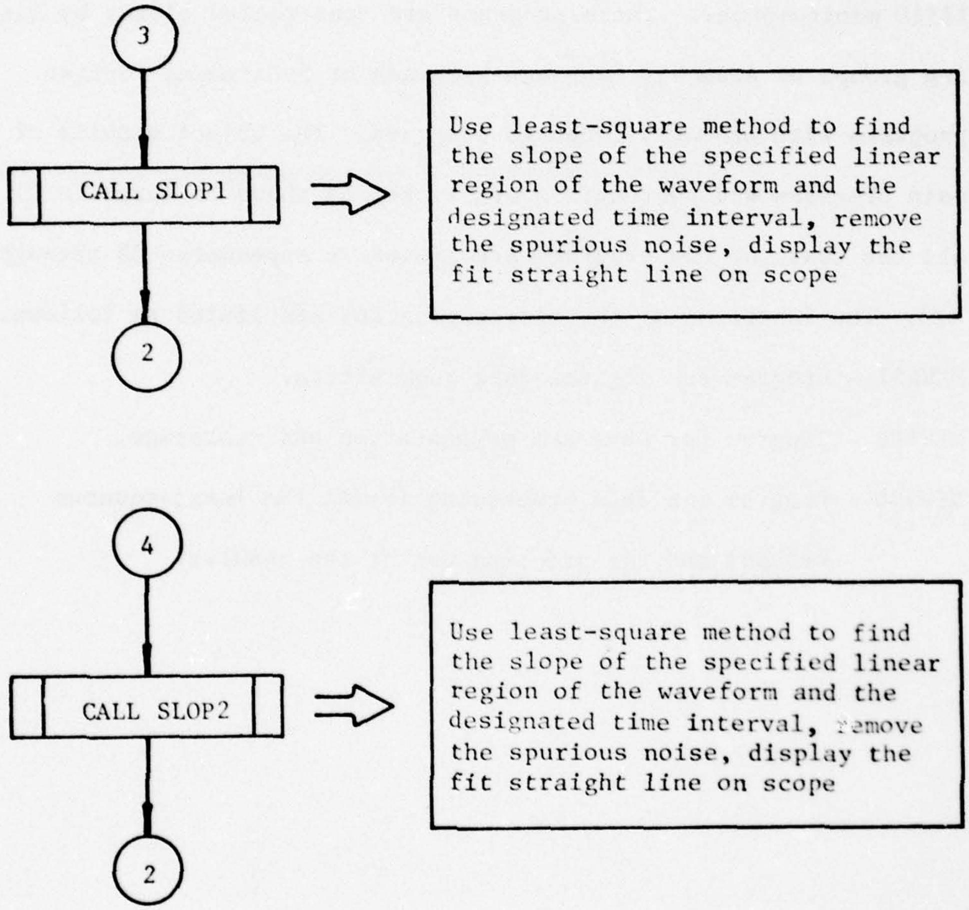
B2. Flow Chart of Waveform Segmentation and Restorage

APPENDIX B



B3. Flow Chart of Data Processing

APPENDIX B



B3. Flow Chart of Data Processing

APPENDIX C

MINICOMPUTER (RT-11) PROGRAMS

This section gives the source programs which are run on the PDP 11/10 minicomputer. These programs are constructed either by linking groups of Assembly Language programs or by linking Fortran programs with Assembly Language programs. The object modules of the main programs and subroutines are linked as shown in Appendix C1, and all the texts of the programs are listed in Appendices C2 through C23. The functions of the source programs are listed as follows:

NEW102 - Program for digital data acquisition.

NEW500 - Program for waveform segmentation and restorage.

NEW900 - Program for data processing (using the least-squares method) and for printing out of the results.

APPENDIX C

```
.R LINK
*NEW102<NEW100,NEW101
```

```
.R LINK
*NEW500<NEW200/C
*NEW101/C
*NEW310,NEW320,NEW330,NEW340/C
*NEW400,NEW420,NEW440/C
*NEW240,NEW250
```

```
.R LINK
*NEW900<NEW802/F/C
*NEW810/O:1/C
*NEW825/O:2/C
*NEW845/O:2/C
*NEW310,NEW320,NEW333,NEW340/O:3/C
*NEW315,NEW325,NEW335,NEW345,NEW355/O:3
```

APPENDIX C

```
        .CSECT
        .TITLE  NEW100.MAC
        .GLOBL  MAIN,TTIN
        LPSCKS=170404
        LPSPB2=170406
        .MCALL  ..V2..,.REGDEF,.FETCH,.ENTER,.WRITE,.CLOSE
        .MCALL  .PRINT,.EXIT
        ..V2..
        .REGDEF
MAIN:   .PRINT  #MSG1               ;PRINT OUT CONTENTS OF MSG1
        JSR     PC,TTIN             ;RECEIVE DATA FROM DECWRITER
        MOV     #DATAIN,R2          ;SET UP R2 WITH ADDRESS
        MOV     (R2)+,BLOCK1        ;MOVE DATA TO BLOCK1
        MOV     (R2)+,CYCLE         ;MOVE DATA TO CYCLE
        CLR     R1                  ;R1=0
        CLR     R2                  ;R2=0
        CLR     R3                  ;R3=0
        CLR     R5                  ;R5=0
        CLR     BLOCK2              ;INITIALIZE BLOCK2 WITH 0
        MOV     #SERV,@#324         ;SET UP ADDRESS AND PRIORITY
        MOV     #340,@#326          ;LEVEL OF INTERRUPT ROUTINE
MULTI:  ADD     #400,R3             ;R3=256*BLOCK1
        INC     R1                  ;INCREASE R1 BY 1
        CMP     BLOCK1,R1           ;R1=BLOCK1?
        BNE     MULTI               ;NO,REPEAT ADDING
        MOV     R3,LREC             ;YES,R3=LREC=256*BLOCK1
        .FETCH  #HNDR,#NAME         ;DEFINE FILE
        .ENTER  #AREA,#1,#NAME,#-1
        .PRINT  #MSG2               ;PRINT OUT CONTENTS OF MSG2
        SUB     BLOCK1,BLOCK2       ;INITIALIZE BLOCK2
        MOV     #BUFF1,R1           ;SET UP R1 WITH ADDRESS
        MOV     R1,R4               ;SET UP R4 WITH ADDRESS
                                    ;SET UP EXTERNAL INTERVAL
                                    ;FROM ZERO BASE, FREQUENCY
                                    ;BEING 100 KHZ, INTERRUPT
                                    ;ENABLE,CLOCK ENABLE
START:  MOV     #1505,LPSCKS        ;WAITING LOOP FOR INTERRUPT
        INC     R2                  ;SET UP INDEX OF REAL-TIME
        CMP     #67130,R2           ;BY THE INSTRUCTION TIME
        BGT     WAIT                ;OF THE WAITING LOOP
        DEC     R5                  ;R5 DECREASE BY 1, REAL-TIME
        CLR     R2                  ;PASSED 23.2*28.248*E-6 SEC
WAIT:   CMP     #0,R3               ;IF R3=0, BUFFER IS FULL
        BLT     START               ;NO, DO NEXT
        CLR     LPSCKS              ;YES,DISABLE THE INTERRUPT
        CMP     R4,#BUFF1           ;DOUBLE-BUFFER METHOD
        BNE     ALPHA               ;R4 POINTS TO BUFF1?
        MOV     #BUFF2,R1           ;YES,R1 POINTS TO BUFF2
```

C2 - NEW100.MAC, Main Program for Data Acquisition

APPENDIX C

```
        BR      STORE
ALPHA:  MOV     #BUFF1,R1           ;NO.R1 POINTS BUFF1
STORE:  ADD     BLOCK1,BLOCK2       ;SET UP BLOCK FOR THE FILE
        MOV     LREC,R3
        .WRITE  #AREA,#1,R4,R3,BLOCK2     ;TRANSFER DATA FROM
        CMP     R4,#BUFF1                 ;MEMORY TO DISK
        BNE     BETA                ;R4 POINTS TO BUFF1?
        MOV     #BUFF2,R4           ;YES,R4 POINTS TO BUFF2
        BR      GAMMA
BETA:   MOV     #BUFF1,R4           ;NO,R4 POINTS TO BUFF1
GAMMA:  .PRINT  #MSG3               ;PRINT OUT CONTENTS OF MSG3
        DEC     CYCLE               ;STORE ONCE
        BLE     FINISH
        JMP     START
SERV:   CMP     #0,R5               ;INTERRUPT SERVICE ROUTINE
        BEQ     OMEGA
        MOV     #-100.,(R1)+        ;SAVE THE COUNT NUMBER AND
        MOV     R5,(R1)+            ;TIMING INDEX IF NECESSARY
        SUB     #2,R3               ;TWO DATA PTS FOR TIMING INDEX
        CLR     R5                  ;RESTORE R5 TO 0
OMEGA:  MOV     LPSPB2,(R1)+        ;SAVE NO. OF COUNTS
        CLR     R2                  ;RESAVE R2 TO 0
        DEC     R3                  ;STORE ONE DATA POINT
        RTI                         ;DISMISS INTERRUPT
FINISH: .PRINT  #MSG4               ;PRINT OUT CONTENTS OF MSG4
        .CLOSE  #1                  ;CLOSE CHANNEL #1
        .EXIT
MSG1:   .ASCIZ/NO.OF BLOCK( <=25 )? NO.OF CYCLES?/
        .EVEN
MSG2:   .ASCIZ/BEGIN THE DATA ACQUISITION!!/
        .EVEN
MSG3:   .ASCIZ/STORE 6400 DATA POINTS/
        .EVEN
MSG4:   .ASCIZ/FINISH!!!/
        .EVEN
BLOCK1: .WORD   0
BLOCK2: .WORD   0
CYCLE:  .WORD   0
LREC:   .WORD   0
AREA:   .BLKW   5
NAME:   .RAD50/DK SHIH50DAT/        ;DATA STORED IN DISK AS THIS
        .EVEN                       ;FILE-NAME
BUFF1:  .BLKW   14400
BUFF2:  .BLKW   14400
HNDR=.
        .CSECT  TELETT
DATAIN: .BLKW   40                  ;DATA TRANSFER FROM 'TTIN'
        .CSECT
        .END    MAIN
```

C2 - NEW100.MAC (Continued)

APPENDIX C

```
        .CSECT
        .TITLE  ASCIZ CODE TO DECIMAL VALUE
        .GLOBL  TTIN
        .MCALL  ..V2..,.REGDEF,.TTYIN
        ..V2..
        .REGDEF
TTIN:   MOV     R0,-(SP)        ;PUSH R0 ON STACK
        MOV     R1,-(SP)        ;PUSH R1 ON STACK
        MOV     R2,-(SP)        ;PUSH R2 ON STACK
        MOV     R3,-(SP)        ;PUSH R3 ON STACK
        MOV     R4,-(SP)        ;PUSH R4 ON STACK
        MOV     R5,-(SP)        ;PUSH R5 ON STACK
        MOV     #TEMP,R2        ;INITIALIZE R2 WITH ADDRESS
INLOOP: .TTYIN  (R2)+           ;INPUT DATA FROM DECWRITER
        BICB    #200,R0         ;IN ASCIZ CODE,CLEAR UPPER BIT
        CMPB    #15,R0          ;'CR' ENDS INPUTTING DATA
        BNE     INLOOP
        MOV     #DATAIN,R4      ;INITIALIZE R4 WITH ADDRESS
        CLR     R1              ;R1=0
DELTA:  CLR     (R4)+           ;CLEAR CONTENTS OF R4
        INC     R1
        CMP     #20,R1          ;CLEAR 20 WORDS
        BNE     DELTA
        MOV     #DATAIN,R4
        CLR     R0
        MOV     #TEMP,R2
        CMPB    (R2),#12        ;A 'LF' LEFT?
        BNE     LOOP0
        INC     R2
LOOP0:  CLR     R1              ;TRANSLATE ASCIZ CODE TO
                                ;DECIMAL NO.AND STORE RESULTS
                                ;IN THE LOCATION OF DATAIN
LOOP1:  CLR     R5
        CLR     R3
        BICB    #300,(R2)       ;CLEAR UPPER TWO BITS
        CMPB    #15,(R2)        ;A 'CR' FROM DECWRITER?
        BNE     AGAIN           ;NO,GET NEXT DATA
        MOV     #100,R0         ;YES,FINISH INPUT
        BR      LOOP3           ;BRANCH TO LOOP3
AGAIN:  CMPB    #54,(R2)        ;A ',' FROM DECWRITER?
        BEQ     LOOP3           ;YES,BRANCH TO LOOP3
        CMP     #0,R1           ;NO,R1=0?
        BNE     LOOP2           ;NO,BRANCH TO LOOP2
        BICB    #60,(R2)        ;NO,SUBSTRACT DATA BY 60
        MOVB    (R2)+,R1        ;MOVE DATA TO R1
        BR      LOOP1           ;BRANCH TO LOOP1
LOOP2:  ADD     R1,R5           ;MULTIPLY BY 10
```

C3 - NEW101.MAC, Conversion of ASCII Codes to Numerical Values

APPENDIX C

```
          INC     R3                ;INCREASE R3 BY 1
          CMP     #12,R3            ;R3=10?
          BNE     LOOP2             ;NO,KEEP ADDING
          BICB    #360,(R2)         ;CLEAR UPPER FOUR BITS
          CLR     R1                ;CLEAR R1
          MOVB    (R2)+,R1          ;MOVE INPUT DATA TO R1
          ADD     R5,R1             ;ADD THE PREVIOUS DATA
          BR      LOOP1             ;BRANCH TO LOOP1
LOOP3:    MOV     R1,(R4)+          ;SAVE RESULT
          CMP     #100,R0           ;FINISH?
          BEQ     OUT               ;YES
          INC     R2                ;INPUT NEXT NUMERICAL NO.
          JMP     LOOP0             ;JMP BACK TO LOOP0
OUT:      MOV     (SP)+,R5          ;POP STACK TO R5
          MOV     (SP)+,R4          ;POP STACK TO R4
          MOV     (SP)+,R3          ;POP STACK TO R3
          MOV     (SP)+,R2          ;POP STACK TO R2
          MOV     (SP)+,R1          ;POP STACK TO R1
          MOV     (SP)+,R0          ;POP STACK TO R0
          RTS     PC                ;RETURN
TEMP:     .BLKW   40
          .EVEN
          .CSECT  TELETT            ;SPECIAL LOCATION FOR
DATAIN:   .BLKW   40                ;RESERVED INPUT DATA
          .CSECT
          .END    TTIN
```

C3 - NEW101.MAC (Continued)

APPENDIX C

```
        .CSECT
        .TITLE  WAVEFORM SEGMENTATION AND RESTORAGE
        .GLOBL  MAIN,TTIN,GRIDY,DISP,CMPR1,CMPR2,CMPR3
        .GLOBL  TIMEX,TIMEY
        .MCALL  ..V2..,.REGDEF,.FETCH,.ENTER,.LOOKUP
        .MCALL  .TTYIN,.PRINT,.READW,.WRITW,.CLOSE,.EXIT
        ..V2..
        .REGDEF
        .MACRO  MULTI   A,B         ;A=256*B
        CLR     R0                  ;EACH BLOCK CONTAINS 256
        CLR     A                   ;WORDS
        ADD     #400,A
        INC     R0
        CMP     B,R0
        BNE     .-14
        .ENDM
MAIN:   .PRINT  #MSG1               ;PRINT OUT CONTENTS OF MSG1
        JSR     PC,TTIN             ;CALL 'TTIN' FOR DATA INPUT
        MOV     #DATAIN,R2          ;SET UP R2 WITH ADDRESS
        MOV     (R2)+,BLOCK1        ;MOVE DATA TO BLOCK1
        .FETCH  #HNDR1,#NAME1       ;DEFINE FILE
        .LOOKUP #AREA1,#1,#NAME1            ;LOOKUP EXISTING FILE
        .FETCH  #HNDR2,#NAME2       ;DEFINE FILE
        .ENTER  #AREA2,#2,#NAME2,#-1        ;ENTER A NEW FILE
        CLR     TIME1               ;SET UP TIMING REFERENCE
        CLR     TIME2               ;EQUALS ZERO
        SUB     #4,BLOCK1           ;INITIALIZE BLOCK1
ALPHA:  ADD     #4,BLOCK1           ;TRANSFER DATA FROM DISK
        .READW  #AREA1,#1,#BUFF30,#2000,BLOCK1
        MOV     #0,FIRST            ;INITIALIZE FOR DISPLAY
        MOV     #2000,LAST          ;INITIALIZE FOR DISPLAY
BETA:   JSR     PC,GRIDY            ;DISPLAY DATA ON SCOPE BY
        JSR     PC,DISP             ;FREQUENCY VERSUS TIME
        .PRINT  #MSG2               ;PRINT MESSAGE
        JSR     PC,TTIN             ;CALL 'TTIN' FOR DATA INPUT
        MOV     #DATAIN,R2          ;SET UP R2 WITH ADDRESS
        MOV     (R2)+,R3            ;MOVE INPUT DATA TO R3
        CMP     R3,#1               ;R3=1?
        BEQ     ALPHA               ;YES,READ NEXT 1024 DATA
        CMP     #4,R3               ;R3<4?
        BLE     DELTA               ;YES,GOTO DELTA
        MOV     (R2)+,FIRST         ;YES,INPUT DATA TO FIRST
        MOV     (R2)+,LAST          ;INPUT DATA TO LAST
        CMP     #2,R3               ;R3=2?
        BEQ     BETA                ;YES,RE-DISPLAY SOME DATA
DELTA:  MOV     #BUFF30,R4          ;SET UP R4 WITH ADDRESS
        MOV     R4,R5               ;R5=R4
        ADD     #4000,R5            ;R5=R5+2048
        ASL     FIRST               ;FIRST=FIRST*2
        ADD     FIRST,R4            ;R4=ADDRESS OF FIRST DATA
```

C4 - NEW200.MAC, Main Program for Waveform Segmentation and Restorage

```
GAMMA:  CMP     (R4),#-100          ;(R4)=-100?
        BNE     PHI                 ;NO,GOTO PHI
        ADD     #2,R4               ;POINT TO NEXT DATA
        MOV     (R4)+,TIME4         ;SAVE DATA IN TIME4
        CMP     R4,R5               ;FINISH?
        BGE     EPSIL               ;YES
        JSR     PC,TIMEX            ;REAL-TIME OF WAITING LOOP
PHI:    MOV     (R4)+,TIME4         ;SAVE DATA IN TIME4
        JSR     PC,TIMEY            ;REAL-TIME OF COUNTS
        CMP     R4,R5               ;FINISH?
        BLT     GAMMA               ;NO,GOTO GAMMA
EPSIL:  CMP     #4,R3               ;YES,R3=4?
        BNE     THETA               ;NO
        JMP     OMEGA               ;YES,READ NEXT 1024 DATA
THETA:  .PRINT  #MSG0               ;PRINT MESSAGE
        JSR     PC,TTIN             ;CALL 'TTIN' FOR INPUT DATA
        MOV     #DATAIN,R2          ;SET UP R2 WITH ADDRESS
        MOV     (R2)+,BLOCK2        ;MOVE INPUT TO BLOCK2
        CMP     #0,BLOCK2           ;BLOCK2=0?
        BEQ     MU                  ;YES
        ADD     #4,BLOCK1           ;NO
        MULTI   LREC2,BLOCK2        ;LREC2=256*BLOCK2
        .READW  #AREA1,#1,#BUFF1,LREC2,BLOCK1    ;READ DATA
        SUB     #4,BLOCK1           ;FROM DISK
        ADD     BLOCK2,BLOCK1       ;SET UP BLOCK1
        MOV     #BUFF1,R4           ;SET UP R4 WITH ADDRESS
        CLR     R0                  ;R0=0
MORE1:  CMP     (R4),#-100          ;(R4)=-100.?
        BNE     MORE2               ;NO
        ADD     #2,R4               ;YES,POINT TO NEXT
        MOV     (R4)+,TIME4         ;SAVE DATA IN TIME4
        JSR     PC,TIMEX            ;REAL-TIME OF WAITING LOOP
        ADD     #2,R0               ;PASS TWO DATA
MORE2:  MOV     (R4)+,TIME4         ;SAVE DATA IN TIME4
        JSR     PC,TIMEY            ;REAL-TIME OF COUNTS
        INC     R0                  ;R0=R0+1
        CMP     R0,LREC2            ;FINISH?
        BLT     MORE1               ;NO
MU:     JMP     ALPHA               ;YES,JUMP BACK
        ;TIME REFERENCE HAS BEEN SET AS 'TIME1' AND 'TIME2'
OMEGA:  .PRINT  #MSG3               ;PRINT MESSAGE
        .PRINT  #MSG4               ;PRINT MESSAGE
        JSR     PC,TTIN             ;CALL 'TTIN' FOR DATA INPUT
        MOV     #DATAIN,R2          ;SET UP R2 WITH ADDRESS
        MOV     (R2)+,BLOCK2        ;MOVE INPUT TO BLOCK2
        MOV     (R2)+,TOTAL         ;MOVE INPUT TO TOTAL
        MOV     (R2)+,BLOCK3        ;MOVE INPUT TO BLOCK3
        MOV     (R2)+,CYCLE1        ;MOVE INPUT TO CYCLE1
        MULTI   LREC1,BLOCK2        ;LREC1=256*BLOCK1
        MULTI   LREC2,BLOCK3        ;LREC2=256*BLOCK3
        MOV     #BUFF1,R4           ;SET UP R4 WITH ADDRESS
```

C4 – NEW200.MAC (Continued)

```
        ;TRANSFER DATA FROM DISK TO MEMORY WITHOUT DISPLAY
        ADD     #4,BLOCK1           ;SET UP NEW BLOCK1
        .READW  #AREA1,#1,R4,LREC1,BLOCK1        ;READ DATA
        MOV     R4,RESER1           ;SET UP RESER1 WITH R4
        MOV     R4,RESER4           ;SER RESER4 WITH R4
        CLR     RESER2
        ;THE FOLLOWING PORTION OF THIS PROGRAM IS TO STORE
        ;DATA BY DEFINITE NUMBER OF BLOCKS
        .PRINT  #MSG5               ;PRINT MESSAGE
        CLR     BLOCK4              ;BLOCK4=0
        SUB     BLOCK3,BLOCK4       ;INITIALIZE BLOCK4
TRY:    JSR     PC,CMPR1            ;FIND BEGINNING PART
        CMP     #200,FLAG           ;END OF FILE?
        BEQ     FINISH              ;YES
        JSR     PC,CMPR2            ;FIND CENTRAL PART
        CMP     #200,FLAG           ;END OF FILE?
        BEQ     FINISH              ;YES
        JSR     PC,CMPR3            ;FIND BEGINNING PART OF NEXT
        CMP     #200,FLAG           ;END OF FILE?
        BEQ     FINISH              ;YES
        ADD     BLOCK3,BLOCK4       ;NO.SET UP BLOCK4
        ;TRANSFER DATA TO THE DISK
        .WRITW  #AREA2,#2,#BUFF3,LREC2,BLOCK4
        .PRINT  #MSG6               ;PRINT MESSAGE
        DEC     CYCLE1              ;SAVE ONE WAVEFORM
        BEQ     FINISH              ;FINISH?
        JMP     TRY                 ;NO.JUMP BACK
FINISH: .PRINT  #MSG7               ;PRINT MESSAGE
        .CLOSE  #1                  ;CLOSE CHANNEL 1
        .CLOSE  #2                  ;CLOSE CHANNEL 2
        .EXIT                       ;FINISH!
MSG0:   .ASCIZ/HOW MANY BLOCKS(=<24) NOT TO BE DISPLAYED?/
        .EVEN
MSG1:   .ASCIZ/WHAT IS 1ST BLOCK OF SHIH50.DAT TO BE READ?/
        .EVEN
MSG2:   .ASCIZ/SKIP(1)?REDISPLAY(2)?LAUNCH,AFTER(3,5)?OK(4)?/
        .EVEN
MSG3:   .ASCIZ/START DATA PROCESSING!/
        .EVEN
MSG4:   .ASCIZ/BLOCK2(=24)? TOTAL ? BLOCK3(=7)? CYCLE1 ?/
        .EVEN
MSG5:   .ASCIZ/SAVE COMPLETE WAVEFORMS FOR PROCESSING!/
        .EVEN
MSG6:   .ASCIZ/TRANSFER ONE CYCLE OF DATA TO DISK!/
        .EVEN
MSG7:   .ASCIZ/FINISH!!/
        .EVEN
NAME1:  .RAD50/DK SHIH50DAT/        ;WHERE DATA STORED BY 100KHZ
                                    ;OF REAL-TIME CLOCK
NAME2:  .RAD50/DK SHIH10DAT/        ;DATA SAVED FOR PROCESSING
CYCLE1: .WORD   0
```

C4 - NEW200.MAC (Continued)

```
BLOCK3: .WORD   0
BLOCK4: .WORD   0
LREC2:  .WORD   0
AREA2:  .BLKW   5
HNDR1:  .BLKW   500
HNDR2:  .BLKW   500
        .CSECT  TELETT              ;SECTION FOR ASCIZ
DATAIN: .BLKW   40                  ;CODE TO DECIMAL
        .CSECT  YDATA               ;SECTION FOR Y-AXIS
YPOS1:  .BLKW   3
YPOS2:  .BLKW   3
YPOS3:  .BLKW   3
YSCALE: .WORD   0
        .CSECT  XDATA               ;SECTION FOR X-AXIS
XPOS1:  .BLKW   3
XPOS2:  .BLKW   3
XPOS3:  .BLKW   3
XSCALE: .WORD
TEST:   .WORD
        .CSECT  NUMBER              ;SECTION FOR NUMERICAL
N0:     .BYTE   76,121,111,105,76          ;CHARACTERS
N1:     .BYTE   0,102,177,100,0
N2:     .BYTE   142,121,11,105,102
N3:     .BYTE   42,101,111,111,66
N4:     .BYTE   30,24,22,177,20
N5:     .BYTE   47,105,105,105,71
N6:     .BYTE   76,111,111,111,62
N7:     .BYTE   101,41,21,11,7
N8:     .BYTE   66,111,111,111,66
N9:     .BYTE   46,111,111,111,76
SPACE:  .BYTE   0,0,0,0,0
        .EVEN
        .CSECT  SCOPE               ;SECTION FOR DISPLAY
BUFF30: .BLKW   2000
FIRST:  .WORD
LAST:   .WORD
        .CSECT  SECOND              ;SECTION FOR TIMING
TIME1:  .WORD   0
TIME2:  .WORD   0
TIME4:  .WORD   0
        .CSECT  TCMPR               ;SECTION FOR WAVEFORM
DATANO: .WORD   0                   ;SEGMENTATION AND STORAGE
AREA1:  .BLKW   5
FLAG:   .WORD   0
LREC1:  .WORD   0
TOTAL:  .WORD   0
BLOCK1: .WORD   0
BLOCK2: .WORD   0
RESER1: .WORD   0
RESER2: .WORD   0
RESER3: .WORD   0
```

C4 – NEW200.MAC (Continued)

```
RESER4: .WORD
BUFF5:  .BLKW   10
BUFF1:  .BLKW   14000
BUFF3:  .BLKW   3500
        .CSECT
        .END    MAIN
```

C4 - NEW200.MAC (Continued)

```
        .CSECT
        .TITLE  NEW250.MAC          ;(TIME1).(TIME2)=0.65535*
        .GLOBL  TIMEX               ;ABS(TIME4)
        .MCALL  ..V2..,.REGDEF
        ..V2..
        .REGDEF
TIMEX.  MOV     R0,-(SP)            ;PUSH R0
        MOV     R1,-(SP)            ;PUSH R1
        MOV     R2,-(SP)            ;PUSH R2
        MOV     R3,-(SP)            ;PUSH R3
        MOV     R4,-(SP)            ;PUSH R4
        MOV     R5,-(SP)            ;PUSH R5
        NEG     TIME4               ;TIME4=-TIME4
        MOV     TIME4,R4            ;SET UP R4 AS TIME4
        CLR     R1                  ;R1=0
        CLR     R2                  ;R2=0
        CMP     #0,R4               ;0<R4?
        BLT     ALPHA               ;YES,
        JMP     OUT                 ;NO,JUMP TO RETURN
ALPHA:  CMP     #50.,R4             ;50<R4?
        BLT     MORE1               ;YES,
BETA:   ADD     #655.,R1            ;R1=655*TIME4
        DEC     R4                  ;R4=R4-1
        CMP     #0,R4               ;R4<0?
        BLT     BETA                ;YES,
        MOV     TIME4,R4            ;SET UP R4
GAMMA:  SUB     #1000.,R1           ;TIME1=R1/1000
        INC     TIME1
        CMP     #1000.,R1
```

C5 - NEW250.MAC, Determination of Real Time from Waiting Loop

```
          BLE     GAMMA
          JMP     ETA
MORE1:    ADD     #655.,R1          ;(R2).(R1)=655*ABS(TIME4)
          ADC     R2                ;DOUBLE-PRECISION
          DEC     R4
          CMP     #0,R4
          BLT     MORE1
MORE2:    SUB     #1000.,R1         ;TIME1=(R2).(R1)/1000
          SBC     R2                ;DOUBLE-PRECISION
          INC     TIME1
          CMP     #0,R2
          BLT     MORE2
          CMP     #1000.,R1
          BLE     MORE2
ETA:      MOV     TIME4,R4          ;SET UP R4
          ADD     R1,TIME2          ;SAVE R1 TO TIME2
          CLR     R1
MORE3:    ADD     #35.,R1           ;R1=35*ABS(TIME4)
          DEC     R4
          CMP     #0,R4
          BLT     MORE3
MORE4:    SUB     #1000.,R1         ;TIME2=(R1)/1000
          INC     TIME2
          CMP     #500.,R1
          BLE     MORE4
OUT:      MOV     (SP)+,R5          ;POP R5
          MOV     (SP)+,R4          ;POP R4
          MOV     (SP)+,R3          ;POP R3
          MOV     (SP)+,R2          ;POP R2
          MOV     (SP)+,R1          ;POP R1
          MOV     (SP)+,R0          ;POP R0
          RTS     PC                ;RETURN
          .CSECT  SECOND            ;SECTION FOR TIMING
TIME1:    .WORD
TIME2:    .WORD
TIME4:    .WORD
          .CSECT
          .END    TIMEX
```

C5 - NEW250.MAC (Continued)

```
         .CSECT
         .TITLE  TIME REFERENCE OF 100K HZ
         .GLOBL  TIMEY
         .MCALL  ..V2...REGDEF
         ..V2..
         .REGDEF
TIMEY:   MOV     R0,-(SP)          ;PUSH R0
         MOV     R1,-(SP)          ;PUSH R1
         MOV     R2,-(SP)          ;PUSH R2
         MOV     R3,-(SP)          ;PUSH R3
         MOV     R4,-(SP)          ;PUSH R4
         MOV     R5,-(SP)          ;PUSH R5
         MOV     TIME4,R5          ;MOVE DATA TO R5
ALPHA:   CLR     R0                ;R0=0
         CMP     #0,R5             ;R5<=0?
         BLE     BETA              ;YES,
         MOV     #328.,R0          ;NO,SET UP R0
         BIC     #100000,R5        ;CLEAR CARRY BIT
BETA:    CMP     #50.,R5           ;50>R5?
         BGT     PHI               ;YES,
GAMMA:   SUB     #100.,R5          ;NO,R0=R5/100
         INC     R0
         CMP     #50.,R5
         BLE     GAMMA
DELTA:   ADD     R0,TIME2          ;ADD R0 TO TIME2
         CMP     #1000.,TIME2      ;1000>TIME2?
         BGT     PHI               ;YES,FINISH.
NU:      SUB     #1000.,TIME2      ;NO,TIME1=TIME2/1000
         INC     TIME1
         CMP     #1000.,TIME2
         BLE     NU                ;TIME=TIME1+TIME2/1000
PHI:     MOV     (SP)+,R5          ;POP R5
         MOV     (SP)+,R4          ;POP R4
         MOV     (SP)+,R3          ;POP R3
         MOV     (SP)+,R2          ;POP R2
         MOV     (SP)+,R1          ;POP R1
         MOV     (SP)+,R0          ;POP R0
         RTS     PC                ;RETURN
         .CSECT  SECOND            ;SECTION FOR TIMING
TIME1:   .WORD
TIME2:   .WORD
TIME4:   .WORD
         .CSECT
         .END    TIMEY
```

C6 - NEW240.MAC, Determination of Real Time from Data

```
        .CSECT
        .TITLE  BEGINNING PORTION
        .GLOBL  CMPR1,TIMEY,TIMEX
        .MCALL  ..V2..,.REGDEF,.READW,.PRINT
        ..V2..
        .REGDEF
        .MACRO  NXBUFF              ;MACRO CALL FOR READING
        .PRINT  #MSG1               ;DATA FROM DISK
        MOV     #BUFF1,R4
        ADD     BLOCK2,BLOCK1
        CMP     TOTAL,BLOCK1        ;TOTAL>BLOCK1?
        BGE     .+6                 ;YES,END OF FILE
        JMP     FIN1
        .READW  #AREA1,#1,R4,LREC1,BLOCK1         ;READ
        CLR     R3                  ;R3=0
        .ENDM
CMPR1:  MOV     R0,-(SP)            ;PUSH R0
        MOV     R1,-(SP)            ;PUSH R1
        MOV     R2,-(SP)            ;PUSH R2
        MOV     R3,-(SP)            ;PUSH R3
        MOV     R4,-(SP)            ;PUSH R4
        MOV     R5,-(SP)            ;PUSH R5
        MOV     RESER1,R4           ;SET UP R4
        MOV     RESER2,R3           ;SET UP R3
        MOV     #12,DATANO          ;SAVE 10 PTS FOR TIMING
        CLR     TEST3               ;TEST3=0
ALPHA:  CLR     TEST1               ;TEST1=0
        CLR     TEST2               ;TEST2=0
BETA:   INC     R3                  ;NEXT DATA POINT
        CMP     R3,LREC1            ;END OF BUFFER?
        BGT     GAMMA               ;YES,GOTO GAMMA
        JMP     EPSIL               ;NO,GOTO EPSIL
GAMMA:  MOV     #BUFF5,R2           ;SET UP R2 WITH ADDRESS
        SUB     #20,R4              ;POINT TO PREVIOUS ADDRESS
        CLR     R5                  ;SAVE 8 DATA POINTS OF
DELTA:  MOV     (R4)+,(R2)+         ;PREVIOUS BUFFER
        INC     R5
        CMP     #10,R5
        BNE     DELTA
        NXBUFF                      ;READ DATA FROM DISK
EPSIL:  CMP     #1,TEST3            ;TEST3=1?
        BEQ     EPSIL1              ;YES,GOTO EPSIL1
        CMP     R4,RESER4           ;R4=RESER4?
        BNE     ZETA                ;NO,GOTO ZETA
        MOV     #1,TEST3            ;TEST3=1
EPSIL1: CMP     (R4),#-100          ;(R4)=100?
        BNE     OMEGA               ;NO,GOTO OMEGA
        MOV     #1,TEST2            ;YES,TEST2=1
        ADD     #2,R3               ;PASS TWO DATA PTS
        ADD     #2,R4               ;POINT TO NEXT PT
```

C7 - NEW400.MAC, Searching for Beginning Portion of Waveform

```
        MOV     (R4)+,TIME4        ;SAVE DATA IN TIME4
        JSR     PC,TIMEX           ;CALL TIMEX
        CMP     R3,LREC1           ;BUFFER IS FULL?
        BLT     OMEGA              ;NO,GOTO OMEGA
        JMP     GAMMA              ;YES,GOTO GAMMA
OMEGA:  MOV     (R4),TIME4         ;GET THE REAL TIME
        JSR     PC,TIMEY           ;CALL TIMEY
ZETA:   CMP     #0,(R4)            ;0<=(R4)?
        BLE     THETA              ;YES,GOTO THETA
        ADD     #2,R4              ;POINT TO NEXT PT
        CMP     #1,TEST2           ;1=TEST2?
        BEQ     RHO                ;YES,GOTO RHO
        MOV     #1,TEST2           ;TEST2=1
        JMP     BETA               ;GOTO BETA
THETA:  CMP     #1,TEST2           ;TEST2=1?
        BEQ     ETA                ;YES,GOTO ETA
        CMP     #1250.,(R4)+       ;FREQ.<80 HZ?
        BLE     PSI                ;YES,GOTO PSI
        JMP     ALPHA              ;NO,GOTO ALPHA
PSI:    MOV     #1,TEST2           ;TEST2=1
        JMP     BETA               ;GOTO BETA
ETA:    CMP     #900.,(R4)+        ;FREQ.<111 HZ?
        BLT     RHO                ;YES,GOTO RHO
        JMP     ALPHA              ;NO,GOTO ALPHA
RHO:    INC     TEST1              ;TEST1=TEST1+1
        CMP     #2,TEST1           ;TEST1=2?
        BEQ     PHI                ;YES,GOTO PHI
        JMP     BETA               ;NO,GOTO BETA
PHI:    MOV     #BUFF3,R1          ;SET UP R1 WITH ADDRESS
        CLR     (R1)+              ;SAVE ONE 0
        MOV     TIME1,(R1)+        ;STORE TIMING DATA
        MOV     TIME2,(R1)+        ;STORE TIMING DATA
        CLR     (R1)+              ;SAVE ONE 0
        MOV     R4,RESER1          ;SAVE R4
        MOV     R3,RESER2          ;SAVE R3
        CMP     R3,#10             ;R3<8?
        BLT     SIGMA              ;YES,GOTO SIGMA
        SUB     #20,R4             ;NO,R4=R4-16
        BR      LAMBDA             ;GOTO LAMBDA
SIGMA:  CLR     R5                 ;R5=0
        MOV     #BUFF5,R2          ;SET UP R2 WITH ADDRESS
KAPPA:  MOV     (R2)+,(R1)+        ;STORE 8 DATA PTS
        INC     DATANO             ;IN TEMPORARY BUFFER
        INC     R5
        CMP     #10,R5
        BNE     KAPPA
        ASL     R3                 ;R3=R3*2
        SUB     R3,R4              ;R4=R4-R3
LAMBDA: MOV     (R4)+,(R1)+        ;STORE DATA POINTS
        INC     DATANO             ;UP TO CURRENT DATA PT
        CMP     R4,RESER1          ;WHICH FITS THE CRITERION
```

C7 - NEW400.MAC (Continued)

```
        BNE     LAMBDA          ;LOOP
        JMP     OUT             ;GOTO OUT
FIN1:   MOV     #200,FLAG       ;SET UP FLAG
OUT:    MOV     R1,RESER3
        MOV     (SP)+,R5        ;POP R5
        MOV     (SP)+,R4        ;POP R4
        MOV     (SP)+,R3        ;POP R3
        MOV     (SP)+,R2        ;POP R2
        MOV     (SP)+,R1        ;POP R1
        MOV     (SP)+,R0        ;POP R0
        RTS     PC              ;RETURN
MSG1:   .ASCIZ/READ AT B/
        .EVEN
TEST1:  .WORD   0
TEST2:  .WORD
TEST3:  .WORD   0
        .CSECT  TCMPR           ;SECTION FOR WAVEFORM
DATANO: .WORD   0               ;SEGMENTATION AND STORAGE
AREA1:  .BLKW   5
FLAG:   .WORD   0
LREC1:  .WORD   0
TOTAL:  .WORD
BLOCK1: .WORD   0
BLOCK2: .WORD   0
RESER1: .WORD   0
RESER2: .WORD   0
RESER3: .WORD   0
RESER4: .WORD
BUFF5:  .BLKW   10
BUFF1:  .BLKW   14000
BUFF3:  .BLKW   3500
        .CSECT  SECOND          ;SECTION FOR TIMING
TIME1:  .WORD
TIME2:  .WORD
TIME4:  .WORD
        .CSECT
        .END    CMPR1
```

C7 - NEW400.MAC (Continued)

```
         .CSECT
         .TITLE  CENTRAL PORTION
         .GLOBL  CMPR2,TIMEY,TIMEX
         .MCALL  ..V2..,.REGDEF,.READW,.PRINT
         ..V2..
         .REGDEF
         .MACRO  NXBUFF              ;MACRO CALL FOR READING
         .PRINT  #MSG1               ;FROM DISK
         MOV     #BUFF1,R4
         ADD     BLOCK2,BLOCK1
         CMP     TOTAL,BLOCK1        ;TOTAL>BLOCK1?
         BGE     .+6                 ;YES,FINISH
         JMP     FIN1
         .READW  #AREA1,#1,R4,LREC1,BLOCK1         ;READ
         CLR     R3
         .ENDM
CMPR2:   MOV     R0,-(SP)            ;PUSH R0
         MOV     R1,-(SP)            ;PUSH R1
         MOV     R2,-(SP)            ;PUSH R2
         MOV     R3,-(SP)            ;PUSH R3
         MOV     R4,-(SP)            ;PUSH R4
         MOV     R5,-(SP)            ;PUSH R5
         MOV     RESER1,R4           ;SAVE R4
         MOV     RESER2,R3           ;SAVE R3
         MOV     RESER3,R1           ;SAVE R1
         MOV     R3,R2               ;R2=R3
         CLR     R0                  ;R0=0
ALPHA:   CLR     R5                  ;R5=0
BETA:    INC     R3                  ;R3=R3+1
         CMP     R3,LREC1            ;BUFFER IS FULL?
         BGT     GAMMA               ;YES,GOTO GAMMA
         JMP     EPSIL               ;NO,GOTO EPSIL
GAMMA:   MOV     RESER2,R3           ;SAVE R3
         MOV     RESER1,R4           ;SAVE R4
DELTA:   MOV     (R4)+,(R1)+         ;STORE DATA POINTS
         INC     DATANO
         INC     R3                  ;R3=R3+1
         CMP     R3,LREC1
         BNE     DELTA               ;LOOP
         NXBUFF                      ;READ DATA FROM DISK
EPSIL:   CMP     (R4),#-100.         ;(R4)=-100?
         BNE     ETA                 ;NO,GOTO ETA
         ADD     #2,R3               ;PASS TWO DATA PTS
         ADD     #2,R4               ;GOTO NEXT DATA PT
         MOV     (R4)+,TIME4         ;SAVE DATA IN TIME4
         JSR     PC,TIMEX            ;CALL TIMEX FOR TIMING

ETA:     MOV     (R4),TIME4          ;SAVE DATA IN TIME4
         JSR     PC,TIMEY            ;CALL TIMEY FOR TIMING

         CMP     #1,R0               ;R0=1?
         BEQ     KAPPA               ;YES,GOTO KAPPA
         CMP     #1030.,(R4)+        ;FREQ.<97 HZ?
         BLE     LAMBDA              ;YES,GOTO LAMBDA
         JMP     ALPHA               ;NO,GOTO ALPHA
KAPPA:   CMP     #1030.,(R4)+        ;FREQ.>97 HZ?
```

C8 - NEW420.MAC, Searching for Central Portion of Waveform

```
        BGE     LAMBDA          ;YES,GOTO LAMBDA
        JMP     ALPHA           ;NO,GOTO ALPHA
LAMBDA: INC     R5              ;R5=R5+1
        CMP     #6,R5           ;R5=6?
        BEQ     MU              ;YES,GOTO MU
        JMP     BETA            ;NO,GOTO BETA
MU:     CMP     #1,R0           ;R0=1?
        BEQ     NU              ;YES,GOTO NU
        MOV     #1,R0           ;NO,R0=1
        JMP     ALPHA           ;GOTO ALPHA
NU:     MOV     R3,RESER2       ;SAVE RESER2
        MOV     R4,RESER1       ;SAVE RESER1
        CMP     RESER3,R1       ;RESER3=R1?
        BNE     TAU             ;NO,GOTO TAU
        SUB     R2,R3           ;R3=R3-R2
TAU:    ASL     R3              ;R3=R3*2
        SUB     R3,R4
SIGMA:  MOV     (R4)+,(R1)+     ;STORE DATA POINTS
        INC     DATANO
        CMP     R4,RESER1       ;R4=RESER1?
        BNE     SIGMA           ;NO,LOOP
        JMP     OUT             ;YES,GOTO OUT
FIN1:   MOV     #200,FLAG       ;SET UP FLAG
OUT:    MOV     R1,RESER3       ;SAVE R1
        MOV     (SP)+,R5        ;POP R5
        MOV     (SP)+,R4        ;POP R4
        MOV     (SP)+,R3        ;POP R3
        MOV     (SP)+,R2        ;POP R2
        MOV     (SP)+,R1        ;POP R1
        MOV     (SP)+,R0        ;POP R0
        RTS     PC              ;RETURN
MSG1:   .ASCIZ/READ AT C/
        .EVEN
        .CSECT  TCMPR           ;SECTION FOR WAVEFORM
DATANO: .WORD   0               ;SEGMENTATION AND STORAGE
AREA1:  .BLKW   5
FLAG:   .WORD   0
LREC1:  .WORD   0
TOTAL:  .WORD
BLOCK1: .WORD   0
BLOCK2: .WORD   0
RESER1: .WORD   0
RESER2: .WORD   0
RESER3: .WORD   0
RESER4: .WORD
BUFF5:  .BLKW   10
BUFF1:  .BLKW   14000
BUFF3:  .BLKW   3500
        .CSECT  SECOND          ;SECTION FOR TIMING
TIME1:  .WORD
TIME2:  .WORD
TIME4:  .WORD
        .CSECT
        .END    CMPR2
```

C8 - NEW420.MAC (Continued)

```
        .CSECT
        .TITLE  FINAL PORTION
        .GLOBL  CMPR3,TIMEY,TIMEX
        .MCALL  ..V2..,.REGDEF,.READW,.PRINT
        ..V2..
        .REGDEF
        .MACRO  NXBUFF              ;MACRO CALL FOR READING
        .PRINT  #MSG1               ;DATA FROM DISK
        MOV     #BUFF1,R4           ;SET UP R4 WITH ADDRESS
        ADD     BLOCK2,BLOCK1
        CMP     TOTAL,BLOCK1        ;TOTAL>BLOCK1?
        BGE     .+6                 ;YES,END OF FILE
        JMP     FIN1
         READW  #AREA1,#1,R4,LREC1,BLOCK1
        CLR     R3
        .ENDM
CMPR3:  MOV     R0,-(SP)            ;PUSH R0
        MOV     R1,-(SP)            ;PUSH R1
        MOV     R2,-(SP)            ;PUSH R2
        MOV     R3,-(SP)            ;PUSH R3
        MOV     R4,-(SP)            ;PUSH R4
        MOV     R5,-(SP)            ;PUSH R5
        MOV     RESER1,R4           ;SAVE R4
        MOV     RESER2,R3           ;SAVE R3
        MOV     RESER3,R1           ;SAVE R1
        MOV     R3,R5               ;R5=R3
ALPHA:  CLR     TEST1               ;TEST1=0
        CLR     TEST2               ;TEST2=0
BETA:   INC     R3                  ;R3=R3+1
        CMP     R3,LREC1            ;BUFFER IS FULL?
        BGT     GAMMA               ;YES,GOTO GAMMA
        JMP     EPSIL               ;NO,GOTO EPSIL
GAMMA:  MOV     RESER2,R3           ;SAVE R3
        MOV     RESER1,R4           ;SAVE R4
DELTA:  MOV     (R4)+,(R1)+         ;STORE DATA POINTS
        INC     DATANO
        INC     R3                  ;R3=R3+1
        CMP     R3,LREC1            ;R3=LREC1?
        BNE     DELTA               ;NO,GOTO DELTA
        CLR     R5
        NXBUFF                      ;YES,READ DATA FROM DISK
EPSIL:  CMP     (R4),#-100          ;(R4)=-100?
        BNE     ZETA                ;NO,GOTO ZETA
        MOV     #1,TEST2            ;TEST2=1
        ADD     #2,R3               ;PASS TWO DATA PTS
        ADD     #2,R4               ;POINT TO NEXT DATA
        MOV     (R4)+,TIME4         ;SAVE DATA IN TIME4
        JSR     PC,TIMEX            ;CALL TIMEX FOR TIMING
ZETA:   MOV     (R4),TIME4          ;SAVE DATA IN TIME4
        JSR     PC,TIMEY            ;CALL TIMEY FOR TIMING
```

C9 - NEW440.MAC Searching for Beginning Portion of Next Waveform

```
        CMP     #0,(R4)         ;0=<(R4)?
        BLE     THETA           ;YES,GOTO THETA
        ADD     #2,R4           ;PASS ONE DATA POINT
        CMP     #1,TEST2        ;TEST2=1?
        BEQ     MU              ;YES,GOTO MU
        MOV     #1,TEST2        ;TEST2=1
        JMP     BETA            ;GOTO BETA
THETA:  CMP     #1,TEST2        ;TEST2=1?
        BEQ     ETA             ;YES,GOTO ETA
        CMP     #1250.,(R4)+    ;FREQ.<80 HZ?
        BLE     XI              ;YES,GOTO XI
        JMP     ALPHA           ;NO,GOTO ALPHA
XI:     MOV     #1,TEST2        ;TEST2=1
        JMP     BETA            ;GOTO BETA
ETA:    CMP     #900.,(R4)+     ;FREQ.<110 HZ?
        BLE     MU              ;YES,GOTO MU
        JMP     ALPHA           ;NO,GOTO ALPHA
MU:     INC     TEST1           ;TEST1=TEST1+1
        CMP     #2,TEST1        ;TEST1=2?
        BEQ     NU              ;YES,GOTO NU
        JMP     BETA            ;NO,GOTO BETA
NU:     MOV     R3,RESER2       ;SAVE RESER2
        MOV     R4,RESER1       ;SAVE RESER1
        MOV     R4,RESER4       ;SAVE RESER4
        SUB     #2,RESER4       ;RESER4=RESER4-2
        CMP     RESER3,R1       ;RESER3=R1?
        BNE     TAU             ;NO,GOTO TAU
        SUB     R5,R3           ;R3=R3-R5
TAU:    ASL     R3              ;R3=R3*2
        SUB     R3,R4           ;R4=R4-R3
OMEGA:  MOV     (R4)+,(R1)+     ;STORE DATA POINTS
        INC     R5
        INC     DATANO
        CMP     R4,RESER1       ;R4=RESER1?
        BNE     OMEGA           ;NO,LOOP
        BR      OUT             ;YES,GOTO OUT
FIN1:   MOV     #200,FLAG       ;SET UP FLAG
OUT:    CLR     (R1)+           ;SAVE ONE 0
        MOV     TIME1,(R1)+     ;SAVE TIMING DATA
        MOV     TIME2,(R1)+     ;SAVE TIMING DATA
        MOV     #BUFF3,R1       ;SET UP R1 WITH ADDRESS
        MOV     DATANO,(R1)+    ;STORE NO. OF DATA PTS
        MOV     RESER2,R3       ;R3=RESER2
        CMP     #10,R3          ;8<=R3?
        BLE     PHI             ;YES,GOTO PHI
        SUB     R3,RESER2       ;RESER2=RESER2-R3
        ASL     R3              ;R3=R3*2
        SUB     R3,RESER1       ;RESER1=RESER1-R3
        BR      PSI             ;GOTO PSI
PHI:    MOV     R5,RESER2       ;SAVE RESER2
        MOV     R4,RESER1       ;SAVE RESER1
```

C9 - NEW440.MAC (Continued)

```
        SUB     #10,RESER2          ;SET UP RESER2
        SUB     #20,RESER1          ;SET UP RESER1
PSI:    MOV     (SP)+,R5            ;POP R5
        MOV     (SP)+,R4            ;POP R4
        MOV     (SP)+,R3            ;POP R3
        MOV     (SP)+,R2            ;POP R2
        MOV     (SP)+,R1            ;POP R1
        MOV     (SP)+,R0            ;POP R0
        RTS     PC                  ;RETURN
MSG1:   .ASCIZ/READ AT E/
        .EVEN
TEST1:  .WORD
TEST2:  .WORD 0
        .CSECT  TCMPR               ;SECTION FOR WAVEFORM
DATANO: .WORD   0                   ;SEGMENTATION AND STORAGE
AREA1:  .BLKW   5
FLAG:   .WORD   0
LREC1:  .WORD   0
TOTAL:  .WORD
BLOCK1: .WORD   0
BLOCK2: .WORD   0
RESER1: .WORD   0
RESER2: .WORD   0
RESER3: .WORD   0
RESER4: .WORD
BUFF5:  .BLKW   10
BUFF1:  .BLKW   14000
BUFF3:  .BLKW   3500
        .CSECT  SECOND              ;SECTION FOR TIMING
TIME1:  .WORD
TIME2:  .WORD
TIME4:  .WORD
        .CSECT
        .END    CMPR3
```

C9 - NEW440,MAC (Continued)

```
        .CSECT
        .TITLE GRIDY,YSHOW          ;DISPLAY GRIDS OF Y-AXIS
        .GLOBL GRIDY,YSHOW          ;AS FREQ.FROM 0 TO 200 HZ
        LPSVC=170416
        .MCALL  ..V2..,.REGDEF
        ..V2..
        .REGDEF
```

C10 - NEW310.MAC, Displaying Vertical Coordinate

```
        .MACRO  SHOW A,B,C,D       ;MACRO CALL FOR DISPLAYING
        MOV     #A,YPOS1           ;DATA IN SUBROUTINE 'YSHOW'
        MOV     #B,YPOS2
        MOV     #C,YPOS3
        MOV     #D,YSCALE
        JSR     PC,YSHOW
        .ENDM
GRIDY:  MOV     R0,-(SP)           ;PUSH R0
        MOV     R1,-(SP)           ;PUSH R1
        MOV     R2,-(SP)           ;PUSH R2
        MOV     R3,-(SP)           ;PUSH R3
        MOV     R4,-(SP)           ;PUSH R4
        MOV     R5,-(SP)           ;PUSH R5
        MOV     #10000,LPSVC       ;ERASE THE SCOPE
        SHOW N2,N0,N0,200.         ;DISPLAY 200 ON SCOPE
        SHOW N1,N8,N0,180.         ;DISPLAY 180 ON SCOPE
        SHOW N1,N6,N0,160.         ;DISPLAY 160 ON SCOPE
        SHOW N1,N4,N0,140.         ;DISPLAY 140 ON SCOPE
        SHOW N1,N2,N0,120.         ;DISPLAY 120 ON SCOPE
        SHOW N1,N0,N0,100.         ;DISPLAY 100 ON SCOPE
        SHOW SPACE,N8,N0,80.       ;DISPLAY 80 ON SCOPE
        SHOW SPACE,N6,N0,60.       ;DISPLAY 60 ON SCOPE
        MOV     (SP)+,R5           ;POP R5
        MOV     (SP)+,R4           ;POP R4
        MOV     (SP)+,R3           ;POP R3
        MOV     (SP)+,R2           ;POP R2
        MOV     (SP)+,R1           ;POP R1
        MOV     (SP)+,R0           ;POP R0
        RTS     PC                 ;RETURN
        .CSECT YDATA     ;SECTION FOR Y-AXIS
YPOS1:  .BLKW   3
YPOS2:  .BLKW   3
YPOS3:  .BLKW   3
YSCALE: .WORD   0
        .CSECT NUMBER              ;SECTION FOR NUMERICAL
N0:     .BYTE 76,121,111,105,76            ;CHARACTERS
N1:     .BYTE 0,102,177,100,0
N2:     .BYTE 142,121,111,105,102
N3:     .BYTE 42,101,111,111,66
N4:     .BYTE 30,24,22,177,20
N5:     .BYTE 47,105,105,105,71
N6:     .BYTE 76,111,111,111,62
N7:     .BYTE 101,41,21,11,7
N8:     .BYTE 66,111,111,111,66
N9:     .BYTE 46,111,111,111,76
SPACE:  .BYTE 0,0,0,0,0
        .EVEN
        .CSECT
        .END    GRIDY
```

C10 - NEW310.MAC (Continued)

```
        .CSECT
        .TITLE YSHOW
        .GLOBL YSHOW
        LPSVC=170416
        LPSVCX=170420
        LPSVCY=170422
        .MCALL  ..V2..,.REGDEF
        ..V2..
        .REGDEF
        ;
        ;THIS SUBROUTINE DISPLAYS NUMERICAL CHARACTERS
        ;ON THE SCOPE AS THE INDEX OF Y-AXIS
        ;
YSHOW:  MOV     R0,-(SP)        ;PUSH R0
        MOV     R1,-(SP)        ;PUSH R1
        MOV     R2,-(SP)        ;PUSH R2
        MOV     R3,-(SP)        ;PUSH R3
        MOV     R4,-(SP)        ;PUSH R4
        MOV     R5,-(SP)        ;PUSH R5
        CLR     R4              ;R4=0
        CLR     TEST            ;INITIALIZE TEST
        CLR     LPSVCX          ;VALUE OF HONRI =0
        CLR     R5              ;R5=0
        CLR     R1              ;R1=0
ALPHA:  ADD     YSCALE,R5       ;R5=FREQ*16
        INC     R1
        CMP     #20,R1
        BNE     ALPHA           ;NO,LOOP
        MOV     R5,YLINE        ;SAVE FREQ*16
        SUB     #40,R5          ;R5=R5-32
        MOV     R5,YSTAR        ;SAVE R5
        MOV     YPOS1,R0        ;MOVE FIRST CHARACTER
        BR      DELTA           ;GOTO DELTA
BETA    MOV     YPOS2,R0        ;MOVE SECOND CHARACTER
        MOV     YSTAR,R5        ;MOVE HONRI. GRID
        BR      DELTA           ;GOTO DELTA
GAMMA:  MOV     YPOS3,R0        ;MOVE THIRD CHARACTER
        MOV     YSTAR,R5        ;MOVE HONRI. GRID
DELTA:  MOV     #-5,R1          ;R1=-5
EPSIL:  ADD     #15,R4          ;R4=R4+13
        MOV     YSTAR,R5        ,MOVE CHARACTER TO R5
        MOV     #-7,R2          ;R2=-7
        MOVB    (R0)+,R3        ;MOVE 8 BITS TO R3
ZETA:   ROLB    R3              ;ROTATE 0 BIT TO CARRY
        BPL     IOTA            ,CARRY BIT=0?
        MOV     #2002,LPSVC     ;NO.SET UP STATUS OF SCOPE
ETA:    TSTB    LPSVC           ;SCOPE READY?
        BPL     ETA             ;NO,WAIT
        MOV     R5,LPSVCY       ,YES,PUT ONE DOT ON SCOPE
        MOV     R4,LPSVCX
```

C11 - NEW320.MAC, Displaying Vertical Coordinate

```
IOTA:   ADD     #11,R5          ;INCREASE VERTI. BY 9
        INC     R2              ;TEST 7 DOTS FOR A COLUMN
        BNE     ZETA
        INC     R1              ;5 COLUMNS FOR A CHARACTER
        BNE     EPSIL
        INC     TEST            ;TEST THREE CHARACTERS
        CMP     #1,TEST         ;TEST=1?
        BEQ     BETA            ;YES,TRY SECOND CHARACTER
        CMP     #2,TEST         ;NO,TEST=2?
        BEQ     GAMMA           ;YES,TRY THIRD CHARACTER
        ADD     #100,R4         ;HONRI. INCREASE BY 64
        MOV     YLINE,R2
KAPPA:  MOV     #2002,LPSVC     ;SET UP STATUS OF SCOPE
MU:     TSTB    LPSVC           ;SCOPE READY?
        BPL     MU              ;NO,WAIT FOR READY
        INC     R4              ;YES,INCREASE HONRI. BY 1
        MOV     R4,LPSVCX       ;GENERATE ONE DOT ON SCOPE
        MOV     R2,LPSVCY       ;KEEP SAME VALUE OF VERTI.
        CMP     #7776,R4        ;OUT OF RANGE OF SCOPE?
        BNE     KAPPA           ;NO,KEEP DISPLAYING
        MOV     (SP)+,R5        ;POP R5
        MOV     (SP)+,R4        ;POP R4
        MOV     (SP)+,R3        ;POP R3
        MOV     (SP)+,R2        ;POP R2
        MOV     (SP)+,R1        ;POP R1
        MOV     (SP)+,R0        ;POP R0
        RTS     PC              ;RETURN
YSTAR:  .WORD   0
YLINE:  .WORD   0
TEST:   .WORD   0
        .CSECT  YDATA           ;SECTION FOR Y-AXIS
YPOS1:  .BLKW   3
YPOS2:  .BLKW   3
YPOS3:  .BLKW   3
YSCALE: .WORD   0
        .CSECT
        .END    YSHOW
```

C11 - NEW320.MAC (Continued)

APPENDIX C

```
        .CSECT
        .TITLE  DISP. THE DATA OF 100K HZ
        .GLOBL  DISP,XSHOW
        LPSVC=170416
        LPSVCX=170420
        LPSVCY=170422
        .MCALL  ..V2..,.REGDEF
        ..V2..
        .REGDEF
        .MACRO  SHOW A,B,C          ;MACRO CALL FOR
        MOV     #A,XPOS1            ;DISPLAYING X-AXIS
        MOV     #B,XPOS2
        MOV     #C,XPOS3
        JSR     PC,XSHOW
        .ENDM
DISP:   MOV     R0,-(SP)            ;PUSH R0
        MOV     R1,-(SP)            ;PUSH R1
        MOV     R2,-(SP)            ;PUSH R2
        MOV     R3,-(SP)            ;PUSH R3
        MOV     R4,-(SP)            ;PUSH R4
        MOV     R5,-(SP)            ;PUSH R5
        MOV     #BUFF30,R0          ;SET UP R0 WITH ADDRESS
        MOV     FIRST,R1            ;SET UP R1
        MOV     LAST,R2             ;SET UP R2
        SUB     R1,R2               ;R2=R2-R1=DATA POINTS
        MOV     R2,DATA1            ;SAVE NO. OF DATA PTS
        MOV     #10000,R4           ;R4=4096
        CLR     R5                  ;R5=0
DEV1:   SUB     R2,R4               ;R5=4096/R2
        INC     R5
        CMP     R2,R4
        BLE     DEV1
        ASL     R1                  ;R1=2*R1
        ADD     R1,R0               ;R0=ADDRESS OF FIRST DATA
        MOV     FIRST,R1            ;R1=NO. OF FIRST DATA
        DEC     R1
START:  CLR     LPSVCX              ;LPSVCX=0
CRT1:   CMP     (R0),#-100.         ;VALUE=-100?
        BNE     ALPHA               ;NO,GOTO ALPHA
        ADD     #4,R0               ;SKIP TWO DATA PTS
        ADD     #2,R1
        MOV     #3900.,TEST         ;DISPLAY IN SPECIAL FORM
        JMP     MOVE1
ALPHA:  MOV     (R0)+,R4            ;SAVE DATA IN R4
        INC     R1                  ;R1=R1+1
        CMP     R1,LAST             ;R1<LAST?
        BLT     CRT2                ;YES,GOTO CRT2
        JMP     OUT                 ;GOTO OUT
CRT2:   MOV     #2002,LPSVC         ;SET UP THE SCOPE
READY:  TSTB    LPSVC               ;SCOPE READY?
```

C12 - NEW 330.MAC, Displaying Data and Horizontal Coordinate

```
          BPL     READY                 ;NO,WAIT
          CMP     #480.,R4              ;480<NO.OF COUNTS?
          BLT     GAMMA0                ;YES,GOTO GAMMA0
          MOV     #3500.,TEST           ;DISPLAY IN SPECIAL FORM
          BR      MOVE1                 ;GOTO MOVE1
GAMMA0:   CMP     #0,R4                 ;0<R4?
          BLT     GAMMA1                ;YES,GOTO GAMMA1
          MOV     #200.,TEST            ;TEST=200
          JMP     MOVE1                 ;GOTO MOVE1
GAMMA1:   MOV     #65000,R2             ;DOUBLE-PRECISION
          MOV     #30,R3
          CLR     TEST
DEV2:     SUB     R4,R2                 ;TEST=1,600,000/R4
          SBC     R3
          INC     TEST
          CMP     #0,R3
          BNE     DEV2
          CMP     R4,R2
          BLE     DEV2
MOVE1:    ADD     R5,LPSVCX             ;DISPLAY A DOT ON SCOPE
          MOV     TEST,LPSVCY
          MOV     LPSVCX,XSCALE
          CMP     DATA1,#150.           ;DATA PTS>150?
          BGT     OK01                  ;YES,GOTO OK01
          CLR     R3
          MOV     R1,R4
OK0.      SUB     #12,R4                ;R4/10=R2
          INC     R3
          CMP     #12,R4
          BLE     OK0
          CMP     #0,R4                 ;R4=0?
          BEQ     OK1                   ;YES,GOTO OK1
          JMP     CRT1                  ;NO,GOTO CRT1
OK1:      SUB     #12,R3                ;R3/10
          CMP     #12,R3
          BLE     OK1
          CMP     #0,R3                 ;R3=0?
          BEQ     OK01                  ;YES,GOTO OK01
          SHOW    SPACE,SPACE,SPACE     ;VERTI. LINE ON SCOPE
          JMP     CRT1                  ;GOTO CRT1
OK01:     CMP     #100.,R1
          BNE     OK11
          SHOW    N1,N0,N0              ;DISPLAY 100 AND VERTI.
          JMP     CRT1                  ;LINE ON SCOPE
OK11:     CMP     #200.,R1
          BNE     OK21
          SHOW    N2,N0,N0              ;DISPLAY 200 AND VERTI.
          JMP     CRT1                  ;LINE ON SCOPE
OK21:     CMP     #300.,R1
          BNE     OK31
          SHOW    N3,N0,N0              ;DISPLAY 300 AND VERTI.
```

C12 - NEW300.MAC (Continued)

```
         JMP     CRT1               ;LINE ON SCOPE
OK31:    CMP     #400.,R1
         BNE     OK41
         SHOW    N4,N0,N0           ;DISPLAY 400 AND VERTI.
         JMP     CRT1               ;LINE ON SCOPE
OK41:    CMP     #500.,R1
         BNE     OK51
         SHOW    N5,N0,N0           ;DISPLAY 500 AND VERTI.
         JMP     CRT1               ;LINE ON SCOPE
OK51:    CMP     #600.,R1
         BNE     OK61
         SHOW    N6,N0,N0           ;DISPLAY 600 AND VERTI.
         JMP     CRT1               ;LINE ON SCOPE
OK61:    CMP     #700.,R1
         BNE     OK71
         SHOW    N7,N0,N0           ;DISPLAY 700 AND VERTI.
         JMP     CRT1               ;LINE ON SCOPE
OK71:    CMP     #800.,R1
         BNE     OK81
         SHOW    N8,N0,N0           ;DISPLAY 800 AND VERTI.
         JMP     CRT1               ;LINE ON SCOPE
OK81:    CMP     #900.,R1
         BNE     OK91
         SHOW    N9,N0,N0           ;DISPLAY 900 AND VERTI.
         JMP     CRT1               ;LINE ON SCOPE
OK91:    CMP     #1000.,R1
         BNE     OK101
         SHOW    N1,N0,N0           ;DISPLAY 100 AND VERTI.
OK101:   JMP     CRT1               ;LINE ON SCOPE
OUT:     MOV     (SP)+,R5           ;POP R5
         MOV     (SP)+,R4           ;POP R4
         MOV     (SP)+,R3           ;POP R3
         MOV     (SP)+,R2           ;POP R2
         MOV     (SP)+,R1           ;POP R1
         MOV     (SP)+,R0           ;POP R0
         RTS     PC                 ;RETURN
DATA1:   .WORD   0
         .CSECT  SCOPE              ;SECTION FOR DATA
BUFF30:  .BLKW   2000
FIRST:   .WORD
LAST:    .WORD
         .CSECT  XDATA              ;SECTION FOR X-VALUE
XPOS1:   .BLKW   3
XPOS2:   .BLKW   3
XPOS3:   .BLKW   3
XSCALE:  .WORD   0
TEST:    .WORD
         .CSECT  NUMBER             ;SECTION FOR NUMERICAL
N0:      .BYTE   76,121,111,105,76        ;CHARACTERS
N1:      .BYTE   0,102,177,100,0
N2:      .BYTE   142,121,111,105,102
```

C12 - NEW330.MAC (Continued)

```
N3:      .BYTE   42,101,111,111,66
N4:      .BYTE   30,24,22,177,20
N5:      .BYTE   47,105,105,105,71
N6:      .BYTE   76,111,111,111,62
N7:      .BYTE   101,41,21,11,7
N8:      .BYTE   66,111,111,111,66
N9:      .BYTE   46,111,111,111,76
SPACE:   .BYTE   0,0,0,0,0
         .EVEN
         .CSECT
         .END    DISP
```

C12 - NEW330.MAC (Continued)

```
         .CSECT
         .TITLE  XSHOW
         .GLOBL  XSHOW
         LPSVC=170416
         LPSVCX=170420
         LPSVCY=170422
         .MCALL  ..V2..,.REGDEF
         ..V2..
         .REGDEF
         ;THIS SUBROUTINE IS TO DISPLAY NUMBERICAL CHARCTERS
         ;ON THE SCOPE AND EXPAND THE HONRI.SCALE
XSHOW:   MOV     R0,-(SP)          ;PUSH R0
         MOV     R1,-(SP)          ;PUSH R1
         MOV     R2,-(SP)          ;PUSH R2
         MOV     R3,-(SP)          ;PUSH R3
         MOV     R4,-(SP)          ;PUSH R4
         MOV     R5,-(SP)          ;PUSH R5
         CLR     R4                ;R4=0
         CLR     LPSVCX            ;LPSVCX=0
         CLR     LPSVCY            ;LPSVCY=0
         MOV     XSCALE,XLINE      ;SAVE XSCALE
         SUB     #36.,XSCALE       ;XSCALE=XSCALE-36
         MOV     XPOS1,R0          ;FIRST CHARACTER
         BR      GAMMA             ;GOTO GAMMA
ALPHA:   MOV     XPOS2,R0          ;2ND CHARACTER
         BR      GAMMA             ;GOTO GAMMA
BETA:    MOV     XPOS3,R0          ;3RD CHARACTER
GAMMA:   MOV     #-5,R1            ;R1=-5 FOR 5 COLUMNS/CHARAC
DELTA:   ADD     #14,XSCALE        ;POSITION FOR NEXT COLUMN
         CLR     R5                ;R5=0
```

C13 - NEW340.MAC, Displaying Horizontal Coordinate

AD-A057 367 TEXAS UNIV AT EL PASO DEPT OF ELECTRICAL ENGINEERING F/G 4/1
A COMPUTERIZED SYSTEM FOR THE REDUCTION OF MIDDLE ATMOSPHERIC E--ETC(U)
JAN 78 S A SHIH, J D MITCHELL DAAD07-74-C-0263
UNCLASSIFIED SP14-77-UA-42 ERADCOM/ASL-CR-78-0263-1 NL

2 OF 2
AD A057 367

END
DATE FILMED
9-78
DDC

APPENDIX C

```
        MOV     #-7,R2          ;R2=-7 FOR 7 ROWS/CHARAC.
        MOVB    (R0)+,R3        ;SAVE DATA IN R3
ZETA:   ROLB    R3              ;ROTATE R3
        BPL     KAPPA           ;CARRY BIT SET,DISPLAY A DOT
        MOV     #2002,LPSVC     ;SET UP SCOPE
PHI:    TSTB    LPSVC           ;SCOPE READY?
        BPL     PHI             ;NO,WAIT
        MOV     R5,LPSVCY       ;YES,PUT A DOT ON SCOPE
        MOV     XSCALE,LPSVCX
KAPPA:  ADD     #11,R5          ;Y-POSITION FOR NEXT DOT
        INC     R2              ;R2=R2+1
        BNE     ZETA            ;FINISH A COLUMN?
        INC     R1              ;R1=R1+1
        BNE     DELTA           ;FINISH A CHARACTER?
        INC     R4              ;R4=R4+1
        CMP     #1,R4           ;R4=1?
        BEQ     ALPHA           ;YES,TRY 2ND CHARACTER
        CMP     #2,R4           ;R4=2?
        BEQ     BETA            ;YES,TRY 3RD CHARACTER.
        MOV     #120.,R5        ;R5=120
MU:     MOV     #2002,LPSVC     ;SET UP THE SCOPE
OMEGA:  TSTB    LPSVC           ;SCOPE READY?
        BPL     OMEGA           ;NO,WAIT
        INC     R5              ;R5=R5+1
        MOV     TEST,R3         ;R3=TEST=Y-VALUE
        SUB     R5,R3           ;R3=R3-R5
        CMP     #300,R3         ;192<R3?
        BLT     PSI             ;YES,GOTO PSI
        ADD     #600,R5         ;R5=R5+384
        MOV     #40000,TEST     ;SET UP TEST
PSI:    MOV     R5,LPSVCY       ;VERTI. STRAIGHT LINE
        MOV     XLINE,LPSVCX
        CMP     #7776,R5        ;4095>=R5?
        BGE     MU              ;YES,GOTO MU
        MOV     (SP)+,R5        ;POP R5
        MOV     (SP)+,R4        ;POP R4
        MOV     (SP)+,R3        ;POP R3
        MOV     (SP)+,R2        ;POP R2
        MOV     (SP)+,R1        ;POP R1
        MOV     (SP)+,R0        ;POP R0
        RTS     PC              ;RETURN
XLINE:  .WORD   0
        .CSECT  XDATA           ;SECTION FOR X-VALUE
XPOS1:  .BLKW   3
XPOS2:  .BLKW   3
XPOS3:  .BLKW   3
XSCALE: .WORD
TEST:   .WORD
        .CSECT
        .END    XSHOW
```

C13 - NEW340.MAC (Continued)

```
C         THIS PROGRAM IS TO GET THE SLOPES CORRESPONDING
C         TO THE POSITIVE AND NEGATIVE CONDUCTIVITIES.
          DIMENSION M(1792),IWNO(100),TIMEO(200),COND(200)
          COMMON M,G,TREF,IFREQ1,IFREQ2,IAXIS,ID1
C         DATA HAS BEEN STORED IN THE DISK UNDER THE
C         FILENAME "SHIH10.DAT"
10        TYPE 30
30        FORMAT(1X,'NO.OF BLOCK?,FIRST AND LAST WAVEFORM ?')
          ACCEPT 50,IBLOCK,IBEGIN,IEND
50        FORMAT(5I6)
          LREC=256*IBLOCK
          IB=IBLOCK*(IBEGIN-2)
          IFLAG3=0
          N=0
          DO 2000 J=IBEGIN,IEND
          IB=IB+IBLOCK
C         SUBROUTINE CHECK IS TO READ DATA FROM DISK
          CALL CHECK(M,IB,LREC)
          ID1=5
          ID2=M(1)-6
          TYPE 100,J,M(1)
100       FORMAT(////1X,'WAVEFORM ',I3,5X,'DATA PTS :',I4)
C         SUBROUTINE GRIDY(MAC) DISPLAYS Y-AXIS ON SCOPE.
150       CALL GRIDY
C         SUBROUTINE DISP(MAC) DISPLAYS DATA POINTS ON SCOPE.
          CALL DISP(M,ID1,ID2,IAXIS)
170       TYPE 180
180       FORMAT(1X,'EXPAND(1)?POS.COND.(2)?NEG.COND.(3)?
        # SKIP(4)?STOP(5)?')
          ACCEPT 50,IFLAG
          GOTO (190,300,400,1950,3000),IFLAG
190       TYPE 200
200       FORMAT(1X,' LIMITS OF THE X-AXIS & Y-AXIS ?')
          ACCEPT 50,ID1,ID2,IFREQ1,IFREQ2
C         EXPANSION OF PART OF THE WAVEFORM
          IFREQ3=INT(3200./(IFREQ2-IFREQ1))
          CALL YEXPD(IFREQ1,IFREQ2,IFREQ3)
C         NO1=100,000/FREQ1,NO2=100,000/FREQ2
          NO1=INT((10000./IFREQ1)*10.+0.5)
          NO2=INT((10000./IFREQ2)*10.+0.5)
C         IEX1=FREQ1*3200/(FREQ2-FREQ1)-400
C         IEX2=250,000/(FREQ2-FREQ1)
          IEX1=INT(IFREQ1*(3200./(IFREQ2-IFREQ1))-400.)
          IEX2=INT((10000./(IFREQ2-IFREQ1))*10.*2.5)
          CALL XEXPD(M,ID1,ID2,NO1,NO2,IEX1,IEX2,IAXIS)
          GOTO 170
300       IF(IFLAG3.EQ.1) GOTO 400
          TYPE 320
320       FORMAT(1X,'##POSITIVE CONDUCTIVITY##')
C         SUBROUTINE SLOP1 IS TO FIND POSITIVE CONDUCTIVITY
          CALL SLOP1
```

C14 - NEW802.FOR, Main Program for Data Processing

APPENDIX C

```
      GOTO 600
400   TYPE 420
420   FORMAT(1X,'##NEGATIVE CONDUCTIVITY##')
      CALL SLOP2
      IFLAG3=1
600   TYPE 700
700   FORMAT(1X,'DISPLAY & CALC.(BOTH=1,ONLY CALC.
     # =2,NO=3)')
      ACCEPT 50,JFLAG
      GOTO (190,300,1000),JFLAG
C     R(CAL)=5*E11,RADII ARE .8275 AND .5"
C     RESPECTIVELY FOR BLUNT PROBE
1000  G1=1.247*G*1E-13
      TYPE 1100,G1,TREF
1100  FORMAT(4X,'CONDUCTIVITY:',E12.4,16X,'TIME :',F9.3)
      N=N+1
      IWNO(N)=J
      TIMEO(N)=TREF
      COND(N)=G1
      GOTO 170
1950  TYPE 1960
1960  FORMAT(1X,'HOW MANY WAVEFORMS NOT TO BE PROCESSED?')
      ACCEPT 50,IK
      IB=IB+IK*IBLOCK
      J=J+IK
      IFLAG3=0
2000  CONTINUE
3000  TYPE 3500
3500  FORMAT(10X,'WAVEFORM',5X,'TIME',20X,'CONDUCTIVITY'
     # /,12X,'(NO)',6X,'(SEC)',23X,'(MHO/CM)'/)
      TYPE 3600,(IWNO(I),TIMEO(I),COND(I),I=1,N)
3600  FORMAT(12X,I3,5X,F9.3,18X,E12.4,/)
4000  STOP
      END
```

C14 - NEW802.FOR (Continued)

```
        .CSECT
        .TITLE  CHECK
        .GLOBL  CHECK
        .MCALL  ..V2..,.REGDEF,.FETCH,.LOOKUP,.READW
        .MCALL  .CLOSE,.EXIT
        ..V2..
        .REGDEF
CHECK:  MOV     R0,-(SP)           ;PUSH R0
        MOV     R1,-(SP)           ;PUSH R1
        MOV     R2,-(SP)           ;PUSH R2
        MOV     R3,-(SP)           ;PUSH R3
        MOV     R4,-(SP)           ;PUSH R4
        MOV     R5,-(SP)           ;PUSH R5
        MOV     2(R5),R1           ;SET UP R1 WITH ADDRESS
        MOV     @4(R5),R2          ;SET UP R2 AS BLOCK
        MOV     @6(R5),R3          ;SET UP R3 AS DATA PTS
        .FETCH  #HNDR,#NAME        ;DEFINE FILE
        .LOOKUP #AREA,#1,#NAME
        .READW  #AREA,#1,R1,R3,R2       ;READ FROM DISK
        .CLOSE  #1                 ;CLOSE CHANNEL #1
        MOV     (SP)+,R5           ;POP R5
        MOV     (SP)+,R4           ;POP R4
        MOV     (SP)+,R3           ;POP R3
        MOV     (SP)+,R2           ;POP R2
        MOV     (SP)+,R1           ;POP R1
        MOV     (SP)+,R0           ;POP R0
        RTS     PC                 ;RETURN
AREA:   .BLKW   5
NAME:   .RAD50/DK SHIH10DAT/       ;DATA STORED AS THIS NAME
        .EVEN
        .CSECT
HNDR=.
        .END
```

C15 - NEW810.MAC, Input of Data from Disk

APPENDIX C

```
      SUBROUTINE SLOP1
      DIMENSION K(1792),X(500),Y(500)
      COMMON K,A,TIME,IFREQ1,IFREQ2,IAXIS,ID1
10    TYPE 30
30    FORMAT(1X,'THE 1ST & LAST PTS.? THE FREQ. DEV.?')
      ACCEPT 50,JS1,JS2,IFDEV
50    FORMAT(3I6)
      JS3=JS1+(JS2-JS1)/2
      N=0
      X(0)=0.0
      TIME=0.
      TIME1=0.
      DO 90 I=8,JS2
      IF(K(I).EQ.0) GOTO 90
      IF(K(I).NE.-100) GOTO 60
      I=I+1
      IF(K(I).GE.0) GOTO 60
      TIME1=(.655)*(ABS(K(I)))*80.
      I=I+1
60    TIME=K(I)/1250.0+TIME1+TIME
      IF(I-JS1) 85,70,75
70    TIME2=TIME
75    N=N+1
      Y(N)=25000.0/K(I)
      L=N-1
      X(N)=K(I)/1250.+X(L)
      IF(I.NE.JS3) GOTO 85
      TIMEF=(X(N)+TIME2)/80.
85    TIME1=0.
90    CONTINUE
      TIME=K(2)+K(3)/1000.+TIMEF
      LOOP=0
      A=0.0
      B=0.0
      GOTO 200
120   TEST=2.5*IFDEV
      GOTO 200
140   TEST=IFDEV
      GOTO 200
160   TEST=0.5*IFDEV
      GOTO 200
180   TEST=0.25*IFDEV
200   NDATA=0
      SUMX=0.0E0
      SUMY=0.0E0
      SUMXX=0.0E0
      SUMXY=0.0E0
      DO 400 I=1,N
      IF(LOOP.EQ.0) GOTO 300
      FREQ=0.25*(.0125*A*X(I)+B)
      DEVI=ABS(Y(I)-FREQ)
```

C16 - NEW825.FOR, Determining Straight Line Fit

```
          IF(DEVI.GT.TEST) GOTO 400
300       SUMX=SUMX+X(I)
          SUMY=SUMY+Y(I)
          SUMXX=SUMXX+X(I)*X(I)
          SUMXY=SUMXY+X(I)*Y(I)
          NDATA=NDATA+1
400       CONTINUE
          D=NDATA*SUMXX-SUMX*SUMX
          C=NDATA*SUMXY-SUMX*SUMY
          E=SUMY*SUMXX-SUMX*SUMXY
          IF(NDATA.LE.1.OR.D.EQ.0.0) GOTO 10
          A=320.0*C/D
          B=4.0*E/D
          LOOP=LOOP+1
          GOTO (120,140,160,180.500),LOOP
500       SUMS=0.0
          DO 600 I=1,N
          FREQ=0.25*(.0125*A*X(I)+B)
          DEVI=ABS(Y(I)-FREQ)
          IF(DEVI.GT.TEST) GOTO 600
          SUMS=SUMS+DEVI**2
600       CONTINUE
          SUMS=SQRT(SUMS/NDATA)
          TYPE 700,NDATF,N,A,B,SUMS,TIME
700       FORMAT(1X,'#RATIO#',I3,'/',I3,2X,'SLOP :',F9.3,2X
        # ,'B=',F7.3,2X,'RES. RMS :',F6.3,2X,'TIME :',F8.3)
          IF(ID1.NE.5) GOTO 760
          GOTO 800
760       IA=INT(5.*(A/(IFREQ2-IFREQ1))*(IAXIS/3.125))
          IB=INT((B-IFREQ1)*(3200./(IFREQ2-IFREQ1))+400.)
780       CALL DSLOP(K,ID1,JS1,IA,IB,IAXIS)
800       RETURN
          END
```

C16 - NEW825.FOR (Continued)

```
      SUBROUTINE SLOP2
      DIMENSION K(1792),X(300),Y(300)
      COMMON K,A,TIME,IFREQ1,IFREQ2,IAXIS,ID1
10    TYPE 30
30    FORMAT(1X,'THE 1ST & LAST PTS.? THE FREQ. DEV.?')
      ACCEPT 50,JS1,JS2,IFDEV
50    FORMAT(3I5)
      JS3=JS1+(JS2-JS1)/2
      N=0
      X(0)=0.0
      TIME=0.
      TIME1=0.
      DO 90 I=8,JS2
      IF(K(I).EQ.0) GOTO 90
      IF(K(I).NE.-100) GOTO 60
      I=I+1
      IF(K(I).GE.0) GOTO 60
      TIME1=(.655)*(ABS(K(I)))*80.
      I=I+1
60    TIME=K(I)/1250.+TIME1+TIME
      IF(I-JS1) 85,70,75
70    TIME2=TIME
75    N=N+1
      Y(N)=25000.0/K(I)
      L=N-1
      X(N)=K(I)/1250.+X(L)
      IF(I.NE.JS3) GOTO 85
      TIMEF=(X(N)+TIME2)/80.
85    TIME1=0.
90    CONTINUE
      TIME=K(2)+K(3)/1000.+TIMEF
      LOOP=0
      A=0.0
      B=0.0
      GOTO 200
120   TEST=2.5*IFDEV
      GOTO 200
140   TEST=IFDEV
      GOTO 200
160   TEST=0.5*IFDEV
      GOTO 200
180   TEST=0.25*IFDEV
200   NDATA=0
      SUMX=0.0E0
      SUMY=0.0E0
      SUMXX=0.0E0
      SUMXY=0.0E0
      DO 400 I=1,N
      IF(LOOP.EQ.0) GOTO 300
      FREQ=0.25*(.0125*A*X(I)+B)
      DEVI=ABS(Y(I)-FREQ)
```

C17 - NEW845.FOR, Determining Straight Line Fit

APPENDIX C

```
      IF(DEVI.GT.TEST) GOTO 400
300   SUMX=SUMX+X(I)
      SUMY=SUMY+Y(I)
      SUMXX=SUMXX+X(I)*X(I)
      SUMXY=SUMXY+X(I)*Y(I)
      NDATA=NDATA+1
400   CONTINUE
      D=NDATA*SUMXX-SUMX*SUMX
      C=NDATA*SUMXY-SUMX*SUMY
      E=SUMY*SUMXX-SUMX*SUMXY
      IF(NDATA.LE.1.OR.D.EQ.0.0) GOTO 10
      A=320.0*C/D
      B=4.0*E/D
      LOOP=LOOP+1
      GOTO (120,140,160,180,500),LOOP
500   SUMS=0.0
      DO 600 I=1,N
      FREQ=0.25*(.0125*A*X(I)+B)
      DEVI=ABS(Y(I)-FREQ)
      IF(DEVI.GT.TEST) GOTO 600
      SUMS=SUMS+DEVI**2
600   CONTINUE
      SUMS=SQRT(SUMS/NDATA)
      TYPE 700,NDATA,N,A,B,SUMS,TIME
700   FORMAT(1X,'#RATIO#',I3,'/',I3,2X,'SLOP :',F9.3,2X
     # ,'B=',F7.3,2X,'RES. RMS :',F6.3,2X,'TIME :',F8.3)
      IF(ID1.NE.5) GOTO 760
      GOTO 790
760   IA=INT(5.*(A/(IFREQ2-IFREQ1))*(IAXIS/3.125))
      IB=INT((B-IFREQ1)*(3200./(IFREQ2-IFREQ1))+400.)
780   CALL DSLOP(K,ID1,JS1,IA,IB,IAXIS)
790   A=-A
800   RETURN
      END
```

C17 - NEW845.FOR (Continued)

APPENDIX C

```
        .CSECT.
        .TITLE  Y-AXIS EXPANSION
        .GLOBL  YEXPD,YSHOW2
        LPSVC=170416
        .MCALL  ..V2..,.REGDEF
        ..V2..
        .REGDEF
        .MACRO  SHOW    A,B,C,D          ;MACRO CALL
        CMP     R1,#D           ;LOWER LIMIT >#D?
        BGT     .+112           ;YES,NOT DISPLAY.
        CMP     R1,#D           ;NO,LOWER LIMIT <#D?
        BLT     .+12            ;YES,COMPARE UPPER LIMIT
        MOV     #1,FLAGY1       ;NO,LOWER LIMIT=#D
        BR      .+40            ;GOTO DISPLAY
        CMP     R2,#D           ;UPPER LIMIT >#D?
        BGE     .+6             ;YES,KEEP TRYING
        JMP     FINISH          ;NO,FINISH!
        CMP     R2,#D           ;UPPER LIMIT =#D?
        BNE     .+12            ;NO,GOTO DISPLAY!
        MOV     #3,FLAGY1       ;UPPER LIMIT = #D
        BR      .+10            ;GOTO DISPLAY
        MOV     #2,FLAGY1       ;#D IS WITHIN LIMITS
        MOV     #A,YPOS1        ;SAVE FIRST CHARACTER
        MOV     #B,YPOS2        ;SAVE SECOND CHARACTER
        MOV     #C,YPOS3        ;SAVE THIRD CHARACTER
        MOV     #D,YSCALE
        JSR     PC,YSHOW2       ;SUBROUTINE FOR DISPLAY
        CLR     FLAGY1
        .ENDM
YEXPD:  MOV     R0,-(SP)        ;PUSH R0
        MOV     R1,-(SP)        ;PUSH R1
        MOV     R2,-(SP)        ;PUSH R2
        MOV     R3,-(SP)        ;PUSH R3
        MOV     R4,-(SP)        ;PUSH R4
        MOV     R5,-(SP)        ;PUSH R5
        MOV     #10000,LPSVC    ;ERASE THE SCOPE
        MOV     @2(R5),FREQ1    ;MOVE LOWER LIMIT
        MOV     @4(R5),FREQ2    ;MOVE UPPER LIMIT
        MOV     @6(R5),FLAGY2   ;VALUE FOR DIVISION ON SCOPE
        CLR     FLAGY1          ;TEST FLAG FOR #D
        MOV     FREQ1,R1        ;SET UP R1
        MOV     FREQ2,R2        ;SET UP R2
        SHOW    SPACE,N6,NO,60.  ;DISPLAY 60 AND HONRI.LINE
        SHOW    SPACE,N7,NO,70.  ;DISPLAY 70 AND HONRI.LINE
        SHOW    SPACE,N8,NO,80.  ;DISPLAY 80 AND HONRI.LINE
        SHOW    SPACE,N9,NO,90.  ;DISPLAY 90 AND HONRI.LINE
        SHOW    N1,NO,NO,100.    ;DISPLAY 100 AND HONRI.LINE
        SHOW    N1,N1,NO,110.    ;DISPLAY 110 AND HONRI.LINE
        SHOW    N1,N2,NO,120.    ;DISPLAY 120 AND HONRI.LINE
        SHOW    N1,N3,NO,130.    ;DISPLAY 130 AND HONRI.LINE
```

C18 - NEW315.MAC, Displaying Vertical Coordinate

```
          SHOW    N1,N4,N0,140        ;DISPLAY 140 AND HONRI.LINE
          SHOW    N1,N5,N0,150        ;DISPLAY 150 AND HONRI.LINE
          SHOW    N1,N6,N0,160        ;DISPLAY 160 AND HONRI LINE
          SHOW    N1,N8,N0,180        ;DISPLAY 180 AND HONRI.LINE
          SHOW    N2,N0,N0,200        ;DISPLAY 200 AND HONRI.LINE
FINISH:   MOV     (SP)+,R5            ;POP R5
          MOV     (SP)+,R4            ;POP R4
          MOV     (SP)+,R3            ;POP R3
          MOV     (SP)+,R2            ;POP R2
          MOV     (SP)+,R1            ;POP R1
          MOV     (SP)+,R0            ;POP R0
          RTS     PC                  ;RETURE
          .CSECT YDATA2               ;SECTION FOR Y-AXIS DISPLAY
YPOS1:    .BLKW   3                   ;AND NUMERICAL CHARACTERS
YPOS2:    .BLKW   3
YPOS3:    .BLKW   3
YSCALE:   .WORD   0
FLAGY1:   .WORD
FLAGY2:   .WORD
FREQ1:    .WORD
FREQ2:    .WORD
          .CSECT NUM1                 ;SECTION FOR NUMERICAL
N0:       .BYTE 76,121,111,105,76     ;CHARACTERS DISPLAY
N1:       .BYTE 0,102,177,100,0
N2:       .BYTE 142,121,111,105,102
N3:       .BYTE 42,101,111,111,66
N4:       .BYTE 30,24,22,177,20
N5:       .BYTE 47,105,105,105,71
N6:       .BYTE 76,111,111,111,62
N7:       .BYTE 101,41,21,11,7
N8:       .BYTE 66,111,111,111,66
N9:       .BYTE 46,111,111,111,76
SPACE:    .BYTE 0,0,0,0,0
          .EVEN
          .CSECT
          .END     YEXPD
```

C18 - NEW315.MAC (Continued)

APPENDIX C

```
        .CSECT
        .TITLE YSHOW2
        .GLOBL YSHOW2
        LPSVC=170416
        LPSVCX=170420
        LPSVCY=170422
        .MCALL   ..V2..,.REGDEF
        ..V2..
        .REGDEF
        ;THIS SUBROUTINE DISPLAYS NUMERICAL CHARCTERS
        ;ON THE SCOPE AND EXPAND THE VERTI. SCALE
YSHOW2: MOV     R0,-(SP)            ;PUSH R0
        MOV     R1,-(SP)            ;PUSH R1
        MOV     R2,-(SP)            ;PUSH R2
        MOV     R3,-(SP)            ;PUSH R3
        MOV     R4,-(SP)            ;PUSH R4
        MOV     R5,-(SP)            ;PUSH R5
        CLR     R4                  ;R4=0
        CLR     TEST                ;TEST=0
        CLR     LPSVCX              ;LPSVCX=0
        CMP     #0,FLAGY1           ;FLAGY1=0?
        BNE     ALPHA0              ;NO,GOTO DISPLAY
        JMP     OUT                 ;YES,GO OUT
ALPHA0: CMP     #1,FLAGY1           ;FLAGY1=0?
        BNE     ALPHA               ;NO,GOTO ALPHA
        MOV     #400.,YLINE         ;YES,LOWER LIMIT
        MOV     YLINE,R5            ;SAVE R5
        BR      GAMMA               ;GOTO GAMMA
ALPHA:  CMP     #3,FLAGY1           ;FLAGY1=3?
        BNE     BETA                ;NO,GOTO BETA
        MOV     #3600.,YLINE        ;YES,UPPER LIMIT
        MOV     YLINE,R5            ;SAVE R5
        BR      GAMMA               ;GOTO GAMMA
BETA:   MOV     YSCALE,R1           ;SET UP R1 WITH ADDRESS
        SUB     FREQ1,R1            ;R1=R1-LOWER LIMIT
        CLR     R5                  ;R5=0
DELTA:  ADD     FLAGY2,R5           ;R5=SCOPE DIVISION*R1
        DEC     R1
        CMP     #0,R1
        BLT     DELTA
        ADD     #400.,R5            ;R5=R5+400(BASE VALUE)
        MOV     R5,YLINE            ;SAVE R5 FOR HONRI.LINE
GAMMA:  SUB     #40,R5              ;R5=R5-32
        MOV     R5,YSTAR            ;DISPLAY 1ST CHARACTER
        MOV     YPOS1,R0
        BR      MU                  ;GOTO MU
OMEGA:  MOV     YPOS2,R0            ;DISPLAY 2ND CHARACTER
        MOV     YSTAR,R5            ;DISPLAY HONRI.LINE
        BR      MU                  ;GOTO MU
THETA:  MOV     YPOS3,R0            ;DISPLAY 3RD CHARACTER
        MOV     YSTAR,R5            ;DISPLAY HONRI.LINE
```

C19 - NEW325.MAC, Displaying Vertical Coordinate

```
MU:     MOV     #-5,R1          ;R1=-5 FOR 5 COLUMNS
IOTA:   ADD     #15,R4          ;POSITION FOR NEXT COLUMN
        MOV     YSTAR,R5        ;INITIALIZE Y-POSITION
        MOV     #-7,R2          ;R2=-7 FOR 7 DOTS/COLUMN
        MOVB    (R0)+,R3        ;SET UP R3 WITH ADDRESS
ETA:    ROLB    R3              ;ROTATE R3
        BPL     NU              ;CARRY BIT SET,DISPLAY A DOT
        MOV     #2002,LPSVC     ;INITIALIZE THE SCOPE
PHI:    TSTB    LPSVC           ;READY?
        BPL     PHI             ;NO,WAIT
        MOV     R5,LPSVCY       ;YES,MOVE Y-VALUE
        MOV     R4,LPSVCX       ;YES,MOVE X-VALUE
NU:     ADD     #11,R5          ;POSITION FOR NEXT DOT
        INC     R2              ;FINISH 7 DOTS?
        BNE     ETA             ;NO,GOTO ETA
        INC     R1              ;FINISH 5 COLUMNS?
        BNE     IOTA            ;NO,GOTO IOTA
        INC     TEST            ;FINISH A CHARACTER
        CMP     #1,TEST
        BEQ     OMEGA           ;TRY 2ND CHARACTER
        CMP     #2,TEST
        BEQ     THETA           ;TRY 3RD CHARACTER
        ADD     #100,R4         ;R4=R4+64 FOR X-POSITION
        MOV     YLINE,R2
SIGMA:  MOV     #2002,LPSVC     ;INITIALIZE THE CSOPE
OK:     TSTB    LPSVC           ;READY?
        BPL     OK              ;NO,WAIT
        INC     R4              ;R4=R4+1 FOR X-POSITION
        MOV     R4,LPSVCX       ;FOR HONRI.LINE
        MOV     R2,LPSVCY       ;THE SAME Y-VALUE
        CMP     #7776,R4        ;4095>=R4?
        BGE     SIGMA           ;YES,GOTO SIGMA
OUT:    MOV     (SP)+,R5        ;POP R5
        MOV     (SP)+,R4        ;POP R4
        MOV     (SP)+,R3        ;POP R3
        MOV     (SP)+,R2        ;POP R2
        MOV     (SP)+,R1        ;POP R1
        MOV     (SP)+,R0        ;POP R0
        RTS     PC              ;RETURN
YSTAR:  .WORD   0
YLINE:  .WORD   0
TEST:   .WORD   0
        .CSECT YDATA2           ;SECTION FOR Y-VALUE AND
YPOS1:  .BLKW   3               ;NUMERICAL CHARACTERS
YPOS2:  .BLKW   3
YPOS3:  .BLKW   3
YSCALE: .WORD   0
FLAGY1: .WORD
FLAGY2: .WORD
FREQ1:  .WORD
FREQ2:  .WORD
        .CSECT
        .END    YSHOW2
```

C19 - NEW 325.MAC (Continued)

APPENDIX C

```
        .CSECT
        .TITLE  DISP. THE DATA OF 100K HZ
        .GLOBL  XEXPD,XSHOW2
        LPSVC=170416
        LPSVCX=170420
        LPSVCY=170422
        .MCALL  ..V2..,.REGDEF
        ..V2..
        .REGDEF
        .MACRO  SHOW A,B,C          ;MACRO CALL FOR VERTI.VALUES
        MOV     #A,XPOS1
        MOV     #B,XPOS2
        MOV     #C,XPOS3
        JSR     PC,XSHOW2
        .ENDM
XEXPD:  MOV     R0,-(SP)            ;PUSH R0
        MOV     R1,-(SP)            ;PUSH R1
        MOV     R2,-(SP)            ;PUSH R2
        MOV     R3,-(SP)            ;PUSH R3
        MOV     R4,-(SP)            ;PUSH R4
        MOV     R5,-(SP)            ;PUSH R5
        MOV     2(R5),R0            ;SET UP R0 WITH ADDRESS
        MOV     @4(R5),R1           ;1ST DATA POINT
        MOV     @6(R5),R2           ;LAST DATA POINT
        MOV     @10(R5),DATA3       ;LOWER LIMIT
        MOV     @12(R5),DATA4       ;UPPER LIMIT
        MOV     @14(R5),X1          ;DISPLAY INDEX
        MOV     @16(R5),Y1          ;DISPLAY INDEX
        MOV     @20(R5),LAST        ;INDEX FOR X-VALUE
        MOV     R1,DATA5            ;R1=DATA5
        MOV     R2,DATA2            ;R2=DATA2
        SUB     R1,R2               ;R2=R2-R1
        MOV     R2,DATA1            ;SAVE R2 AS DATA POINTS
        CLR     R3                  ;R3=0
DEV1:   SUB     #3,R2               ;R3=R2/3
        INC     R3
        CMP     #3,R2
        BLE     DEV1
        MOV     R3,LAST             ;LAST=R3
OMEGA:  ASL     R1                  ;R1=2*R1
        ADD     R1,R0               ;R0=ADDRESS OF 1ST DATA PT
        DEC     DATA5               ;FIRST=FIRST-1
START:  CLR     LPSVCX              ;LPSVCX=0
CRT1:   CMP     (R0),#-100          ;VALUE=-100?
        BNE     ALPHA               ;NO,GOTO ALPHA
        ADD     #4,R0               ;SKIP TWO DATA PTS
        ADD     #2,DATA5
        ADD     #100.,LPSVCX        ;DISPLAY IN SPECIAL FORM
        MOV     #3900.,R3
        JMP     MOVE3               ;GOTO MOVE3
```

C20 - NEW335.MAC, Displaying Data and Horizontal coordinate

```
ALPHA:  MOV     (R0)+,R4            ;SAVE DATA IN R4
        INC     DATA5               ;DATA5=DATA5+1
        CMP     DATA5,DATA2         ;DATA5<=DATA2?
        BLE     CRT2                ;YES,GOTO CRT2
        JMP     OUT                 ;NO,GOTO OUT
CRT2:   MOV     #2002,LPSVC         ;SET UP STATUS OF SCOPE
READY:  TSTB    LPSVC               ;SCOPE READY?
        BPL     READY               ;NO,WAIT
        CMP     R4,DATA3            ;FREQ.>LOWER LIMIT?
        BLE     GAMMA               ;YES,GOTO GAMMA
        MOV     #250.,R3            ;NO,MOVE 250 TO Y-VALUE
        BR      MOVE1               ;GOTO MOVE1
GAMMA:  CMP     R4,DATA4            ;FREQ.<UPPER LIMIT?
        BGE     DELTA               ;YES,GOTO DELTA
        MOV     #3850.,R3           ;MOVE 3850 TO Y-VALUE
        BR      MOVE1               ;GOTO MOVE1
DELTA:  CLR     R2                  ;R2=0
        CLR     R3                  ;R3=0
        MOV     R4,RESER1           ;SAVE R4
        CLR     R4                  ;R4=0
        MOV     Y1,R1               ;SET UP R1
THETA:  ADD     R1,R2               ;DOUBLE-PRECISION
        ADC     R3                  ;(R2)(R3)=1280*R1
        INC     R4
        CMP     #1280.,R4
        BGE     THETA
        MOV     RESER1,R4           ;SAVE R4=NO.OF COUNTS
        CLR     R1                  ;R1=0
ETA:    SUB     R4,R2               ;DOUBLE-PRECISION
        SBC     R3                  ;R1=(R2)(R3)/R4
        INC     R1
        CMP     #0,R3
        BLT     ETA
        CMP     R4,R2
        BLE     ETA
        SUB     X1,R1               ;R1=R1-X1
        MOV     R1,R3               ;R3=R1
MOVE1:  CLR     R1                  ;R1=0
        MOV     LAST,R2
DEV3:   SUB     R2,R4               ;R1=R4/R2
        INC     R1
        CMP     R2,R4
        BLE     DEV3
        ADD     R1,LPSVCX           ;ADD R1 TO X-VALUE
MOVE3:  MOV     R3,LPSVCY           ;R3=Y-VALUE ON SCOPE
        MOV     R3,TEST
        MOV     LPSVCX,XSCALE       ;SAVE LPEVCX
        MOV     DATA5,R1
        CMP     DATA1,#150.         ;DATA PTS>150?
        BGT     OK01                ;YES,GOTO OK01
        CLR     R2
```

C20 - NEW335.MAC (Continued)

```
        MOV     R1,R4
OK0:    SUB     #12,R4          ;R4/10=R2
        INC     R2
        CMP     #12,R4
        BLE     OK0
        CMP     #0,R4           ;R4=0?
        BEQ     OK1             ;YES,GOTO OK1
        JMP     CRT1            ;NO,GOTO CRT1
OK1:    SUB     #12,R2          ;R2/10
        CMP     #12,R2
        BLE     OK1
        CMP     #0,R2           ;R2=0?
        BEQ     OK01            ;YES,GOTO OK01
        SHOW    SPACE,SPACE,SPACE  ;VERTI.LINE ON SCOPE
        JMP     CRT1            ;GOTO CRT1
OK01:   CMP     #100.,R1
        BNE     OK11
        SHOW    N1,N0,N0        ;DISPLAY 100 AND VERTI.
        JMP     CRT1            ;LINE ON SCOPE
OK11:   CMP     #200.,R1
        BNE     OK21
        SHOW    N2,N0,N0        ;DISPLAY 200 AND VERTI.
        JMP     CRT1            ;LINE ON SCOPE
OK21    CMP     #300.,R1
        BNE     OK31
        SHOW    N3,N0,N0        ;DISPLAY 300 AND VERTI.
        JMP     CRT1            ;LINE ON SCOPE
OK31:   CMP     #400.,R1
        BNE     OK41
        SHOW    N4,N0,N0        ;DISPLAY 400 AND VERTI.
        JMP     CRT1            ;LINE ON SCOPE
OK41:   CMP     #500.,R1
        BNE     OK51
        SHOW    N5,N0,N0        ;DISPLAY 500 AND VERTI.
        JMP     CRT1            ;LINE ON SCOPE
OK51:   CMP     #600.,R1
        BNE     OK61
        SHOW    N6,N0,N0        ;DISPLAY 600 AND VERTI.
        JMP     CRT1            ;LINE ON SCOPE
OK61:   CMP     #700.,R1
        BNE     OK71
        SHOW    N7,N0,N0        ;DISPLAY 700 AND VERTI.
        JMP     CRT1            ;LINE ON SCOPE
OK71.   CMP     #800.,R1
        BNE     OK81
        SHOW    N8,N0,N0        ;DISPLAY 800 AND VERTI.
        JMP     CRT1            ;LINE ON SCOPE
OK81:   CMP     #900.,R1
        BNE     OK91
        SHOW    N9,N0,N0        ;DISPLAY 900 AND VERTI.
        JMP     CRT1            ;LINE ON SCOPE
```

C20 - NEW335.MAC (Continued)

APPENDIX C

```
OK91:     CMP     #1000.,R1
          BNE     OK101
          SHOW    N1,N0,N0            ;DISPLAY 100 AND VERTI.
          JMP     CRT1                ;LINE ON SCOPE
OK101:    CMP     #1400.,R1
          BNE     OK111
          SHOW    N1,N4,N0            ;DISPLAY 140 AND VERTI.
OK111:    JMP     CRT1                ;LINE ON SCOPE
OUT:      MOV     LAST,@20(R5)
          MOV     (SP)+,R5            ;POP R5
          MOV     (SP)+,R4            ;POP R4
          MOV     (SP)+,R3            ;POP R3
          MOV     (SP)+,R2            ;POP R2
          MOV     (SP)+,R1            ;POP R1
          MOV     (SP)+,R0            ;POP R0
          RTS     PC                  ;RETURN
DATA1:    .WORD   0
DATA2:    .WORD
DATA3:    .WORD
DATA4:    .WORD
DATA5:    .WORD   0
X1:       .WORD
Y1:       .WORD
RESER1:   .WORD
RESER2:   .WORD
LAST:     .WORD
          .CSECT  XDATA2              ;SECTION FOR X-VALUE
XPOS1:    .BLKW   3
XPOS2:    .BLKW   3
XPOS3:    .BLKW   3
XSCALE:   .WORD   0
TEST:     .WORD
          .CSECT  NUM1                ;SECTION FOR NUMERICAL
N0:       .BYTE   76,121,111,105,76         ;CHARACTERS
N1:       .BYTE   0,102,177,100,0
N2:       .BYTE   142,121,111,105,102
N3:       .BYTE   42,101,111,111,66
N4:       .BYTE   30,24,22,177,20
N5:       .BYTE   47,105,105,105,71
N6:       .BYTE   76,111,111,111,62
N7:       .BYTE   101,41,21,11,7
N8:       .BYTE   66,111,111,111,66
N9:       .BYTE   46,111,111,111,76
SPACE:    .BYTE   0,0,0,0,0
          .EVEN
          .CSECT
          .END    XEXPD
```

C20 - NEW335.MAC (Continued)

```
        .CSECT
        .TITLE  XSHOW2
        .GLOBL  XSHOW2
        LPSVC=170416
        LPSVCX=170420
        LPSVCY=170422
        .MCALL  ..V2..,.REGDEF
        ..V2..
        .REGDEF
        ;
        ;THIS SUBROUTINE IS TO DISPLAY NUMBERICAL CHARCTERS
        ;ON THE SCOPE AND EXPAND THE HONRI.SCALE
        ;
XSHOW2: MOV     R0,-(SP)        ;PUSH R0
        MOV     R1,-(SP)        ;PUSH R1
        MOV     R2,-(SP)        ;PUSH R2
        MOV     R3,-(SP)        ;PUSH R3
        MOV     R4,-(SP)        ;PUSH R4
        MOV     R5,-(SP)        ;PUSH R5
        CLR     R4              ;R4=0
        CLR     LPSVCX          ;LPSVCX=0
        CLR     LPSVCY          ;LPSVCY=0
        MOV     XSCALE,XLINE    ;SAVE XSCALE
        SUB     #36.,XSCALE     ;XSCALE=XSCALE-36
        MOV     XPOS1,R0        ;FIRST CHARACTER
        BR      GAMMA           ;GOTO GAMMA
ALPHA:  MOV     XPOS2,R0        ;2ND CHARACTER
        BR      GAMMA           ;GOTO GAMMA
BETA:   MOV     XPOS3,R0        ;3RD CHARACTER
GAMMA:  MOV     #-5,R1          ;R1=-5 FOR 5 COLUMNS/CHARAC.
DELTA:  ADD     #14,XSCALE      ;POSITION FOR NEXT COLUMN
        CLR     R5              ;R5=0
        MOV     #-7,R2          ;R2=-7 FOR 7 ROWS/CHARAC.
        MOVB    (R0)+,R3        ;SAVE DATA IN R3
ZETA:   ROLB    R3              ;ROTATE R3
        BPL     KAPPA           ;CARRY BIT SET,DISPLAY A DOT
        MOV     #2002,LPSVC     ;SET UP SCOPE
PHI:    TSTB    LPSVC           ;SCOPE READY?
        BPL     PHI             ;NO,WAIT
        MOV     R5,LPSVCY       ;YES,PUT A DOT ON SCOPE
        MOV     XSCALE,LPSVCX
KAPPA.  ADD     #11,R5          ;Y-POSITION FOR NEXT DOT
        INC     R2              ;R2=R2+1
        BNE     ZETA            ;FINISH A COLUMN?
        INC     R1              ;R1=R1+1
        BNE     DELTA           ;FINISH A CHARACTER?
        INC     R4              ;R4=R4+1
        CMP     #1,R4           ;R4=1?
        BEQ     ALPHA           ;YES,TRY 2ND CHARACTER
        CMP     #2,R4           ;R4=2?
        BEQ     BETA            ;YES,TRY 3RD CHARACTER.
```

C21 - NEW345.MAC, Displaying Vertical Coordinate

```
        MOV     #120.,R5         ;R5=120
MU:     MOV     #2002,LPSVC      ;SET UP THE SCOPE
OMEGA:  TSTB    LPSVC            ;SCOPE READY?
        BPL     OMEGA            ;NO,WAIT
        INC     R5               ;R5=R5+1
        MOV     TEST,R3          ;R3=TEST=Y-VALUE
        SUB     R5,R3            ;R3=R3-R5
        CMP     #300,R3          ;192<R3?
        BLT     PSI              ;YES,GOTO PSI
        ADD     #600,R5          ;R5=R5+384
        MOV     #40000,TEST      ;SET UP TEST
PSI:    MOV     R5,LPSVCY        ;VERTI.STRAIGHT LINE
        MOV     XLINE,LPSVCX
        CMP     #7776,R5         ;4095>=R5?
        BGE     MU               ;YES,GOTO MU
        MOV     (SP)+,R5         ;POP R5
        MOV     (SP)+,R4         ;POP R4
        MOV     (SP)+,R3         ;POP R3
        MOV     (SP)+,R2         ;POP R2
        MOV     (SP)+,R1         ;POP R1
        MOV     (SP)+,R0         ;POP R0
        RTS     PC               ;RETURN
XLINE:  .WORD   0
        .CSECT  XDATA2           ;SECTION FOR X-VALUE
XPOS1:  .BLKW   3
XPOS2:  .BLKW   3
XPOS3:  .BLKW   3
XSCALE: .WORD
TEST:   .WORD
        .CSECT
        .END    XSHOW2
```

```
        .CSECT
        .TITLE  DSLOP           ;DISPLAY A FITTED STRAIGHT
        .GLOBL  DSLOP           ;LINE ON SCOPE
        LPSVC=170416
        LPSVCX=170420
        LPSVCY=170422
        .MCALL  ..V2..,.REGDEF
        ..V2..
        .REGDEF
DSLOP:  MOV     R0,-(SP)        ;PUSH R0
        MOV     R1,-(SP)        ;PUSH R1
        MOV     R2,-(SP)        ;PUSH R2
        MOV     R3,-(SP)        ;PUSH R3
        MOV     R4,-(SP)        ;PUSH R4
        MOV     R5,-(SP)        ;PUSH R5
        MOV     2(R5),R0        ;SET UP R0 WITH ADDRESS
        MOV     @4(R5),FIRST    ;FIRST=S
        MOV     @6(R5),R4       ;SET UP 1ST DATA POINT
        MOV     @10(R5),IA      ;IA=SLOPE
        MOV     @12(R5),IB      ;IB=INTERCEPT
        MOV     @14(R5),LAST    ;LAST=DISPLAY INDEX
        CLR     R3              ;R3=0
        SUB     FIRST,R4        ;R4=R4-S
        MOV     FIRST,R2        ;R2=FIRST
        ASL     R2              ;R2=R2*2
        ADD     R2,R0           ;ADDRESS OF 1ST DATA PT
        MOV     LAST,R2         ;SAVE R2
OK00:   MOV     (R0)+,R1        ;MOVE DATA TO R1
OK01:   SUB     R2,R1           ;R1/R2=R3
        INC     R3
        CMP     R2,R1
        BLE     OK01
        DEC     R4              ;R4=R4-1
        CMP     #0,R4           ;0<=R4?
        BLE     OK00            ;YES,KEEP DOING
        MOV     R3,VCX          ;X-VALUE OF 1ST DATA PT
        CLR     LPSVCX          ;LPVCX=0
        CLR     LPSVCY          ;LPVCY=0
        CLR     R0              ;R0=0
        MOV     VCX,R4          ;SAVE R4
        MOV     IA,R3           ;R3=IA
        MOV     IB,R2           ;R2=IB
OK35:   MOV     #2002,LPSVC     ;SET UP THE SCOPE
OK4:    TSTB    LPSVC           ;SCOPE READY?
        BPL     OK4             ;NO,WAIT
        CMP     #0,R0           ;YES,R0=0?
        BNE     BETA            ;NO,GOTO BETA
        SUB     R3,R2           ;YES,R2=R2-R3
        SUB     #50 ,R4         ;R4=R4-50
        CMP     #0,R2           ;0>=R2?
```

C22 - NEW355.MAC, Displaying Straight Line Fit

```
         BGE     GAMMA               ;YES,GOTO GAMMA
         CMP     #0,R4               ;0<R4?
         BLT     OMEGA               ;YES,GOTO OMAGA
GAMMA:   MOV     VCX,R4              ;SAVE X-POSITION
         MOV     IA,R3               ;SAVE SLOPE
         MOV     IB,R2               ;SAVE INTERCEPT
         MOV     #1,R0               ;R0=1
         BR      OK4                 ;GOTO OK4
BETA:    ADD     R3,R2               ;R2=R2+R3
         ADD     #50.,R4             ;R4=R4+50
         CMP     #7776,R4            ;4095<=R4?
         BLE     OUT                 ;YES,GOTO OUT
         CMP     #7776,R2            ;4095<=R2?
         BLE     OUT                 ;YES,GOTO OUT
OMEGA:   MOV     R4,LPSVCX           ;DISPLAY A DOT
         MOV     R2,LPSVCY
         BR      OK4                 ;GOTO OK4
OUT:     MOV     (SP)+,R5            ;POP R5
         MOV     (SP)+,R4            ;POP R4
         MOV     (SP)+,R3            ;POP R3
         MOV     (SP)+,R2            ;POP R2
         MOV     (SP)+,R1            ;POP R1
         MOV     (SP)+,R0            ;POP R0
         RTS     PC                  ;RETURN
VCX:     .WORD   0
IA:      .WORD   0
IB:      .WORD   0
FIRST:   .WORD
LAST:    .WORD
         .END
```

C22 - NEW355.MAC (Continued)

```
        .CSECT
        .TITLE  DISP  THE DATA OF 100K HZ
        .GLOBL  DISP,XSHOW
        LPSVC=170416
        LPSVCX=170420
        LPSVCY=170422
        .MCALL  ..V2..,.REGDEF
        ..V2..
        .REGDEF
        .MACRO  SHOW A,B,C          ;MACRO CALL FOR
        MOV     #A,XPOS1            ;DISPLAYING X-AXIS
        MOV     #B,XPOS2
        MOV     #C,XPOS3
        JSR     PC,XSHOW
        .ENDM
DISP:   MOV     R0,-(SP)            ;PUSH R0
        MOV     R1,-(SP)            ;PUSH R1
        MOV     R2,-(SP)            ;PUSH R2
        MOV     R3,-(SP)            ;PUSH R3
        MOV     R4,-(SP)            ;PUSH R4
        MOV     R5,-(SP)            ;PUSH R5
        MOV     2(R5),R0            ;SET UP R0 WITH ADDRESS
        MOV     @4(R5),R1           ;SET UP R1
        MOV     @6(R5),R2           ;SET UP R2
        MOV     @10(R5),RESER1      ;SET UP RESER1
        MOV     R1,FIRST            ;SAVE R1
        MOV     R2,DATA2            ;SAVE R2
        SUB     R1,R2               ;R2=R2-R1=DATA POINTS
        MOV     R2,DATA1            ;SAVE NO. OF DATA PTS
        CLR     R5                  ;R5=0
DEV1:   SUB     #4,R2               ;R5=R2/4
        INC     R5
        CMP     #4,R2
        BLE     DEV1
        MOV     R5,LAST             ;SAVE R5
OMEGA:  ASL     R1                  ;R1=2*R1
        ADD     R1,R0               ;R0=ADDRESS OF FIRST DATA
        MOV     FIRST,R1            ;R1=NO. OF FIRST DATA
        DEC     R1
START:  CLR     LPSVCX              ;LPSVCX=0
CRT1:   CMP     (R0),#-100.         ;VALUE=-100?
        BNE     ALPHA               ;NO,GOTO ALPHA
        ADD     #4,R0               ;SKIP TWO DATA PTS
        ADD     #2,R1
        ADD     #100.,LPSVCX        ;LPSVCX=LPSVCX+100
        MOV     #3900.,R5           ;DISPLAY IN SPECIAL FORM
        JMP     MOVE3
ALPHA:  MOV     (R0)+,R4            ;SAVE DATA IN R4
        INC     R1                  ;R1=R1+1
        CMP     R1,DATA2            ;R1<DATA2?
```

C23 - NEW333.MAC, Displaying Data and Horizontal Coordinate

```
        BLT     CRT2            ;YES,GOTO CRT2
        JMP     OUT             ;GOTO OUT
CRT2:   MOV     #2002,LPSVC     ;SET UP THE SCOPE
READY:  .TSTB   LPSVC           ;SCOPE READY?
        BPL     READY           ;NO,WAIT
        CMP     #480.,R4        ;480<NO. OF COUNTS?
        BLT     GAMMA0          ;YES,GOTO GAMMA0
        MOV     #3500.,R5       ;DISPLAY IN SPECIAL FORM
        BR      MOVE1           ;GOTO MOVE1
GAMMA0: CMP     #0,R4           ;0<R4?
        BLT     GAMMA1          ;YES,GOTO GAMMA1
        MOV     #200.,R5        ;R5=200
        ADD     #50.,LPSVCX
        JMP     MOVE3           ;GOTO MOVE3
GAMMA1: CMP     #2000.,R4
        BGT     GAMMA2
        MOV     #300.,R5
        ADD     #7,LPSVCX
        JMP     MOVE3
GAMMA2: MOV     #65000,R2       ;DOUBLE-PRECISION
        MOV     #30,R3
        CLR     R5
DEV2:   SUB     R4,R2           ;R5=1,600.000/R4
        SBC     R3
        INC     R5
        CMP     #0,R3
        BNE     DEV2
        CMP     R4,R2
        BLE     DEV2
MOVE1:  CLR     R3              ;R3=0
        MOV     LAST,R2         ;SET UP R2
DEV3:   SUB     R2,R4
        INC     R3
        CMP     R2,R4
        BLE     DEV3
        ADD     R3,LPSVCX
MOVE3:  MOV     R5,LPSVCY       ;DISPLAY A DOT ON SCOPE
        MOV     R5,TEST
        MOV     LPSVCX,XSCALE
        CMP     DATA1,#150.     ;DATA PTS>150?
        BGT     OK01            ;YES,GOTO OK01
        CLR     R3
        MOV     R1,R4
OK0:    SUB     #12,R4          ;R4/10=R2
        INC     R3
        CMP     #12,R4
        BLE     OK0
        CMP     #0,R4           ;R4=0?
        BEQ     OK1             ;YES,GOTO OK1
        JMP     CRT1            ;NO,GOTO CRT1
OK1:    SUB     #12,R3          ;R3/10
```

C23 - NEW333.MAC (Continued)

APPENDIX C

```
          CMP     #12,R3
          BLE     OK1
          CMP     #0,R3               ;R3=0?
          BEQ     OK01                ;YES,GOTO OK01
          SHOW    SPACE,SPACE,SPACE   ;VERTI. LINE ON SCOPE
          JMP     CRT1                ;GOTO CRT1
OK01:     CMP     #100.,R1
          BNE     OK11
          SHOW    N1,N0,N0            ;DISPLAY 100 AND VERTI.
          JMP     CRT1                ;LINE ON SCOPE
OK11:     CMP     #200.,R1
          BNE     OK21
          SHOW    N2,N0,N0            ;DISPLAY 200 AND VERTI.
          JMP     CRT1                ;LINE ON SCOPE
OK21:     CMP     #300.,R1
          BNE     OK31
          SHOW    N3,N0,N0            ;DISPLAY 300 AND VERTI.
          JMP     CRT1                ;LINE ON SCOPE
OK31:     CMP     #400.,R1
          BNE     OK41
          SHOW    N4,N0,N0            ;DISPLAY 400 AND VERTI.
          JMP     CRT1                ;LINE ON SCOPE
OK41:     CMP     #500.,R1
          BNE     OK51
          SHOW    N5,N0,N0            ;DISPLAY 500 AND VERTI.
          JMP     CRT1                ;LINE ON SCOPE
OK51:     CMP     #600.,R1
          BNE     OK61
          SHOW    N6,N0,N0            ;DISPLAY 600 AND VERTI.
          JMP     CRT1                ;LINE ON SCOPE
OK61:     CMP     #700.,R1
          BNE     OK71
          SHOW    N7,N0,N0            ;DISPLAY 700 AND VERTI.
          JMP     CRT1                ;LINE ON SCOPE
OK71:     CMP     #800.,R1
          BNE     OK81
          SHOW    N8,N0,N0            ;DISPLAY 800 AND VERTI.
          JMP     CRT1                ;LINE ON SCOPE
OK81:     CMP     #900.,R1
          BNE     OK91
          SHOW    N9,N0,N0            ;DISPLAY 900 AND VERTI.
          JMP     CRT1                ;LINE ON SCOPE
OK91:     CMP     #1000.,R1
          BNE     OK101
          SHOW    N1,N0,N0            ;DISPLAY 100 AND VERTI
OK101:    JMP     CRT1                ;LINE ON SCOPE
OUT:      MOV     LAST,RESER1         ;SAVE RESER1
          MOV     (SP)+,R5            ;POP R5
          MOV     (SP)+,R4            ;POP R4
          MOV     (SP)+,R3            ;POP R3
          MOV     (SP)+,R2            ;POP R2
```

C23 - NEW333.MAC (Continued)

```
        MOV     (SP)+,R1        ;POP R1
        MOV     (SP)+,R0        ;POP R0
        RTS     PC              ;RETURN
RESER1: .WORD
DATA1:  .WORD   0
DATA2:  .WORD
        .CSECT  SCOPE           ;SECTION FOR DATA
FIRST:  .WORD
LAST:   .WORD
        .CSECT  XDATA           ;SECTION FOR X-VALUE
XPOS1:  .BLKW   3
XPOS2:  .BLKW   3
XPOS3:  .BLKW   3
XSCALE: .WORD   0
TEST:   .WORD
        .CSECT  NUMBER          ;SECTION FOR NUMERICAL
N0:     .BYTE   76,121,111,105,76       ;CHARACTERS
N1:     .BYTE   0,102,177,100,0
N2:     .BYTE   142,121,111,105,102
N3:     .BYTE   42,101,111,111,66
N4:     .BYTE   30,24,22,177,20
N5:     .BYTE   47,105,105,105,71
N6:     .BYTE   76,111,111,111,62
N7:     .BYTE   101,41,21,11,7
N8:     .BYTE   66,111,111,111,66
N9:     .BYTE   46,111,111,111,76
SPACE:  .BYTE   0,0,0,0,0
        .EVEN
        .CSECT
        .END    DISP
```

C23 - NEW333.MAC (Continued)

☆ U.S. GOVERNMENT PRINTING OFFICE: 1978 777-093/27